

FABIO LOUVATTI DO CARMO

**Melhoria da Convergência do Método  
ICA-MAP para Remoção de Ruído em Sinal de  
Voz**

**Brasil**

**2013**

FABIO LOUVATTI DO CARMO

## **Melhoria da Convergência do Método ICA-MAP para Remoção de Ruído em Sinal de Voz**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Universidade Federal do Espírito Santo

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Evandro Ottoni Teatini Salles

Brasil

2013

---

FABIO LOUVATTI DO CARMO

Melhoria da Convergência do Método ICA-MAP para Remoção de Ruído em  
Sinal de Voz/ FABIO LOUVATTI DO CARMO. – Brasil, 2013-  
55 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Evandro Ottoni Teatini Salles

Dissertação (mestrado) – Universidade Federal do Espírito Santo  
Centro Tecnológico  
Programa de Pós-Graduação em Engenharia Elétrica, 2013.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III.  
Faculdade de xxx. IV. Título

CDU 02:141:005.7

---


**FABIO LOUVATTI DO CARMO**

**Melhoria da Convergência do Método ICA-MAP para  
Remoção de Ruído em Sinal de Voz**

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau Mestre em Engenharia Elétrica.

Aprovada em 10 de dezembro de 2013.

**COMISSÃO EXAMINADORA**



Prof. Dr. Evandro Ottoni Teatini Salles  
Universidade Federal do Espírito Santo  
Orientador



Prof. Dr. Mário Sarcinelli Filho  
PPGEE/DEL - Universidade Federal do Espírito Santo



Prof. Dr. Klaus Fabian Côco  
DEL - Universidade Federal do Espírito Santo

*Dedico este trabalho aos meus pais, Manoel Marinho e Zuleiga, pela educação que me deram, e à minha esposa Alini, por todo apoio.*

# Agradecimentos

Agradeço ao amigo e orientador Dr. Evandro Ottoni Teatini Salles por toda ajuda e suporte, ao CNPq pela bolsa de estudos, aos amigos e familiares pelo carinho e palavras de incentivo. E principalmente a Deus por ter me dado paciência em muitos momentos, força para prosseguir e iluminado o caminho.

# Resumo

O problema de separação de fontes consiste em recuperar um sinal latente de um conjunto de misturas observáveis. Em problemas de *denoising*, que podem ser encarados como um problema de separação de fontes, é necessário extrair um sinal de voz não observado a partir de um sinal contaminado por ruído. Em tal caso, uma importante abordagem baseia-se na análise de componentes independentes (modelos ICA). Neste sentido, o uso da ICA com o algoritmo *maximum a posteriori* (MAP) é conhecido como ICA-MAP. O emprego de duas transformações individuais para sinal de voz e ruído pode proporcionar uma melhor estimativa dentro de um ambiente linear. Esse trabalho apresenta uma modificação feita no algoritmo ICA-MAP a fim de melhorar sua convergência. Foi observado, através de testes, que é possível limitar a magnitude do vetor gradiente, usado para estimar os parâmetros do modelo de *denoising*, e assim melhorar a estabilidade do algoritmo. Tal adaptação pode ser entendida como uma restrição no problema de otimização original. Outra abordagem proposta é aproximar a derivada do modelo GGM (*generalized gaussian model*) em torno de zero por uma *spline*. Para acelerar o algoritmo, é aplicado um passo variável no algoritmo do gradiente. Testes comparativos foram realizados empregando-se bases padrões de dados de voz (masculino e feminino) e de ruído. No final, os resultados obtidos são comparados com técnicas clássicas, a fim de destacar as vantagens do método.

**Palavras-chaves:** Melhoramento de Voz, Maximum a posteriori (MAP), Análise das Componentes Independentes (ICA), Ruído Não-Gaussiano, Otimização

# Abstract

Source separation consists on recovering a latent signal from a set of observable mixtures. In denoising problems, that could be faced as a source separation problem, it is necessary to extract an unobserved signal of voice contaminated by noise. In such a case, an important approach is based on independent component analysis (ICA models). Especially, the use of ICA with the maximum a posteriori (MAP) estimation is known as ICA-MAP and, unlike other approaches, employs two different transformations, one for voice signal and another for noise, providing a better estimate within a linear environment. This work presents a modification to the algorithm ICA-MAP in order to improve its convergence. It has been observed through tests that it is possible to limit the magnitude of the gradient vector, used for estimating the parameters of the denoising model, and thus improving the stability of the algorithm. Such adaptation can be understood as a restriction of the original optimization problem. Another proposed approach is to approximate the derivative of the model GGM (generalized gaussian model) around zero by using a spline function. In order to accelerate the algorithm, it is applied a variable step in the ascent gradient algorithm. Comparative tests were conducted employing standardized voice (male and female) and noise databases. At the end, the results are compared with classical techniques in order to highlight the advantages of the method.

**Key-words:** Speech Enhancement, Maximum a posteriori (MAP), Independent Component Analysis (ICA), Non-Gaussian Noise, Optimization.



# Lista de ilustrações

Figura 1 – Diagrama de blocos do algoritmo de melhoramento de voz ICA-MAP. . . . .	15
Figura 2 – Ilustração do problema <i>cocktail-party</i> . . . . .	17
Figura 3 – Exemplo do <i>cocktail-party</i> . As fontes originais $s_j$ , as misturas $x_i$ e as componentes independentes estimadas. . . . .	18
Figura 4 – Branqueamento. Adaptado de (FUKUNAGA, 1990) . . . . .	19
Figura 5 – Exemplo de PCA bidimensional. (a) Dados originais $\mathbf{x}$ , e (b) as componentes principais $\mathbf{y}$ . . . . .	20
Figura 6 – Exemplo de ICA. . . . .	26
Figura 7 – Gráficos da solução exata (estimativa) e suas aproximações. . . . .	34
Figura 8 – Exemplos do modelo GGM com alguns valores de $\beta$ . . . . .	35
Figura 9 – Distribuição <i>a priori</i> de $\beta$ . . . . .	36
Figura 10 – (a) Logaritmo natural do GGM com $\beta = 2$ . (b) Derivada de (a). . . . .	37
Figura 11 – (a) Evolução de uma componente de $\mathbf{x}$ no algoritmo ICA-MAP(gen). (b) Evolução do gradiente em (3.5). . . . .	38
Figura 12 – (a) Evolução de uma componente de $\mathbf{x}$ no algoritmo ICA-MAP(gen) com limitação do gradiente. (b) Evolução do gradiente em (3.5) com a limitação em (3.29). . . . .	39
Figura 13 – (a) Evolução de uma componente de $\mathbf{x}$ no algoritmo ICA-MAP(gen) com limitação do gradiente e com passo variável. (b) Evolução do gradiente em (3.5) com a limitação em (3.29) e passo variável em (3.31). . . . .	40
Figura 14 – Valores de $\lambda_{it}$ calculados no decorrer das iterações usando limitação do gradiente. . . . .	41
Figura 15 – (a) Exemplo de interpolação da derivada do GGM por <i>spline</i> Hermite cúbica em torno de zero. (b) <i>Zoom</i> de (a) na fronteira do intervalo de interpolação. . . . .	42
Figura 16 – (a) Evolução de uma componente de $\mathbf{x}$ no algoritmo ICA-MAP(gen) com interpolação do gradiente por <i>spline</i> . (b) Evolução do gradiente em (3.5) com a interpolação do gradiente. . . . .	42
Figura 17 – Valores de $\lambda_{it}$ calculados no decorrer das iterações usando interpolação por <i>spline</i> . . . . .	43
Figura 18 – Verificação da esparsidade das componentes independentes. . . . .	45
Figura 19 – (a) Sinal de voz masculina limpo e seu espectrograma. (b) Sinal contaminado com ruído gaussiano em 0 dB e seu espectrograma. (c) Estimativa do sinal limpo com subtração espectral. (d) Estimativa do sinal limpo com <i>Sparse Code Shrinkage</i> . (e) Estimativa do sinal limpo com ICAMAP 1. (f) Estimativa do sinal limpo com ICAMAP 2. . . . .	50

Figura 20 – (a) Sinal de voz masculina limpo e seu espectrograma. (b) Sinal contaminado com ruído não gaussiano em 0 dB e seu espectrograma. (c) Estimativa do sinal limpo com subtração espectral. (d) Estimativa do sinal limpo com *Sparse Code Shrinkage*. (e) Estimativa do sinal limpo com ICAMAP 1. (f) Estimativa do sinal limpo com ICAMAP 2. . . . . 51

# Lista de tabelas

Tabela 1 – Comparação de SNR (em dB) e SD de sinais corrompidos com 0 dB por ruídos gaussianos e não-gaussianos. . . . .	48
Tabela 2 – Comparação de SNR (em dB) e SD de sinais corrompidos com 5 dB por ruídos gaussianos e não-gaussianos. . . . .	49

# Lista de símbolos

$x$	Sinal limpo
$y$	Sinal corrompido por ruído
$v$	Ruído
$s$	Componente independente
$p$	Função densidade de probabilidade
$A$	Matriz de mistura
$W$	Matriz de separação
$\delta$	Desvio padrão
$\sigma$	Variância
$m$	Média
$C$	Matriz de covariância
$E$	Operador valor esperado
$\mathbf{I}$	Matriz identidade
$\mathbf{w}$	Vetor de pesos
$V$	Matriz de branqueamento
$\alpha$	Momento
$\mu$	Momento central
$\kappa$	<i>Kurtosis</i> normalizada
$\psi$	Função característica
$\phi$	Logaritmo natural da função característica

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Motivação	13
1.2	Problema	14
1.3	Objetivo	15
1.4	Publicação	15
1.5	Divisão da dissertação	16
<b>2</b>	<b>ANÁLISE DE COMPONENTES INDEPENDENTES</b>	<b>17</b>
2.1	Introdução	17
2.2	Análise de componentes independentes	17
2.3	Análise das Componentes Principais e Branqueamento	18
2.4	Estatísticas de alta ordem	21
2.5	Definição de ICA	23
2.5.1	Restrições do ICA	24
2.5.2	Ambiguidades do ICA	25
2.5.3	Ilustração do ICA	26
2.6	Algoritmo fastICA	27
2.6.1	O algoritmo de ponto fixo fastICA	28
<b>3</b>	<b>ICA-MAP</b>	<b>30</b>
3.1	Introdução	30
3.2	A abordagem ICA-MAP	30
3.2.1	Estimativa MAP	30
3.2.2	Casos Especiais	31
3.2.3	Modelo Gaussiano Generalizado	35
3.3	Modificação Proposta	36
<b>4</b>	<b>TESTES E RESULTADOS</b>	<b>44</b>
4.1	Metodologia para realização dos testes	44
4.2	Testes realizados	46
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>52</b>
	Referências	53

# 1 Introdução

## 1.1 Motivação

A voz é um dos mais importantes meios de comunicação humana. O sinal de voz é gerado por um locutor e pode ser transmitido por vários meios, que pode ser um cabo de fibra óptica, fios de cobre ou até mesmo o ar, até chegar ao sistema auditivo do ouvinte. Durante este percurso, o sinal de voz pode ser corrompido por vários tipos de ruído. Então, em algumas aplicações, é desejável utilizar alguma técnica que remova este ruído. Este tipo de técnica é conhecida como melhoramento de voz. Melhoramento de voz visa aumentar a qualidade perceptual e a inteligibilidade do sinal. O objetivo é utilizar algoritmos para reduzir o nível do fundo ruidoso em sinais de voz capturados por um microfone, sem afetar sobremaneira o conteúdo do sinal relativo exclusivamente à voz. Em geral, assume-se que o sinal  $y$ , capturado pelo microfone, é uma versão do sinal de voz  $x$ , corrompida por uma fonte ruidosa  $v$ . Ou seja, pode-se escrever que

$$y = x + v. \quad (1.1)$$

Para melhoramento do sinal de voz, utilizam-se técnicas de processamento digital de sinais.

Em (BOLL, 1979), modificada em (EPHRAIM; MALAH, 1984), é apresentada a técnica denominada Subtração Espectral, muito utilizada para remoção de fundo ruidoso aditivo em sinal de voz em canal único. Subtração Espectral utiliza amplitude espectral de tempo curto (STSA) e estima o sinal de voz limpo a partir da subtração da STSA do ruído, mantendo a fase de tempo curto do sinal corrompido. Na subtração espectral, a STSA é estimada como a raiz quadrada do estimador *maximum likelihood* (ML, ou estimador de máxima verossimilhança) de cada variância da componente espectral do sinal. Em sistemas baseados na filtragem de *Wiener*, a STSA é obtida baseada no estimador do erro médio quadrático mínimo (MMSE) ótimo de cada componente espectral do sinal (LIM; OPPENHEIM, 1979), (SOON; KOH, 2000). Ambos estimadores supracitados são obtidos supondo densidade de probabilidade gaussiana de ambos sinais, de voz e ruído.

Em (HYVÄRINEN, 1999), (LEE; LEWICKI, 2000) e (POTAMITIS; FAKOTAKIS; KOKKINAKIS, 2001), foi proposto o uso de algoritmos *maximum a posteriori* (MAP). E muitas vezes, a estimação é acompanhada de uma transformação linear. Estudos apresentaram o uso da transformada discreta de cosseno (DCT) e transformada Karhunen-Loève (KLT) (GAZOR; ZHANG, 2005), transformada *wavelet* (XIE; PIERCE; ULABY, 2002) e análise de componentes independentes linear (ICA linear) (HYVÄRINEN, 1999) (LEE; LEWICKI, 2000) (POTAMITIS; FAKOTAKIS; KOKKINAKIS, 2001) (HONG; ROSCA; BALAN, 2004). Diferentemente de ML, algoritmos de estimação MAP utilizam

a informação *a priori* do sinal de voz. Tipicamente, o modelo usado em sinais de voz é assumido ser não gaussiano. Observou-se que a modelagem não gaussiana leva a uma regra de estimação não linear, e foi avaliado que se obtém melhores resultados em muitos casos (BREITHAUPT; MARTIN, 2003).

Sinais de voz podem ser bem descritos pela distribuição laplaciana, que foi empregada em (HONG; ROSCA; BALAN, 2004) e (GAZOR; ZHANG, 2005). O modelo de misturas gaussianas (GMM) foi usado em (XIE; PIERCE; ULABY, 2002). Em (HYVÄRINEN, 1999), os sinais de voz foram modelados por uma combinação de distribuições levemente super-gaussianas, Laplacianas e fortemente super-gaussianas, e o algoritmo obtido é referenciado como *Sparse Code Shrinkage* (SCS). Pode-se assumir que os dados de voz admitem esparsidade, e a ICA linear pode ser usada para extrair dados esparsos. O algoritmo em (HONG; ROSCA; BALAN, 2004) considera a existência de uma possível correlação entre as componentes do ruído no domínio ICA.

Os algoritmos de melhoramento mencionados acima assumem que o ruído tem distribuição gaussiana, e com isso a transformação linear aplicada nos dados não afeta a distribuição do ruído. Em (ZOU *et al.*, 2008), o ruído é assumido ser não gaussiano, e é mostrado que, utilizando-se o algoritmo MAP, podem ser empregadas transformações ICA individuais para o sinal limpo e ruído, e assim obter melhores estimativas do sinal de voz limpo. Tal algoritmo é conhecido como ICA-MAP. Também é proposto o uso do modelo gaussiano generalizado (GGM), originalmente introduzido em (BOX; TIAO, 1973), que é um modelo flexível para modelar uma ampla faixa de distribuições não gaussianas.

A Figura 1.1 mostra o diagrama de blocos de uma aplicação do ICA-MAP com o uso de detecção de atividade de voz (VAD) como em (JEANNES; FAUCON, 1994) e (RAMIREZ *et al.*, 2004). Na aplicação, a VAD pode ser usada para silenciar o canal de áudio na ausência de voz, e também para reestimar os parâmetros de treino do algoritmo num processo adaptativo. Entretanto, neste trabalho apenas o algoritmo MAP será avaliado, sem o uso da VAD.

## 1.2 Problema

A partir do modelo apresentado em (1.1), o problema que permeia esse trabalho consiste em estimar o sinal latente  $x$  a partir da observação de  $y$ , sob o ponto de vista de separação de fontes, por meio da abordagem ICA. Em especial, nesse trabalho, serão mostradas modificações no algoritmo ICA-MAP, proposto por (ZOU *et al.*, 2008). Notou-se que existem alguns problemas na convergência do algoritmo, devido a algumas características da densidade de probabilidade dos dados. Sabe-se que a ICA, como técnica de extração de características, retorna uma transformação linear que leva os dados para um espaço esparsos (HYVÄRINEN; HOYER; OJA, 1999). Com isso, a função densidade

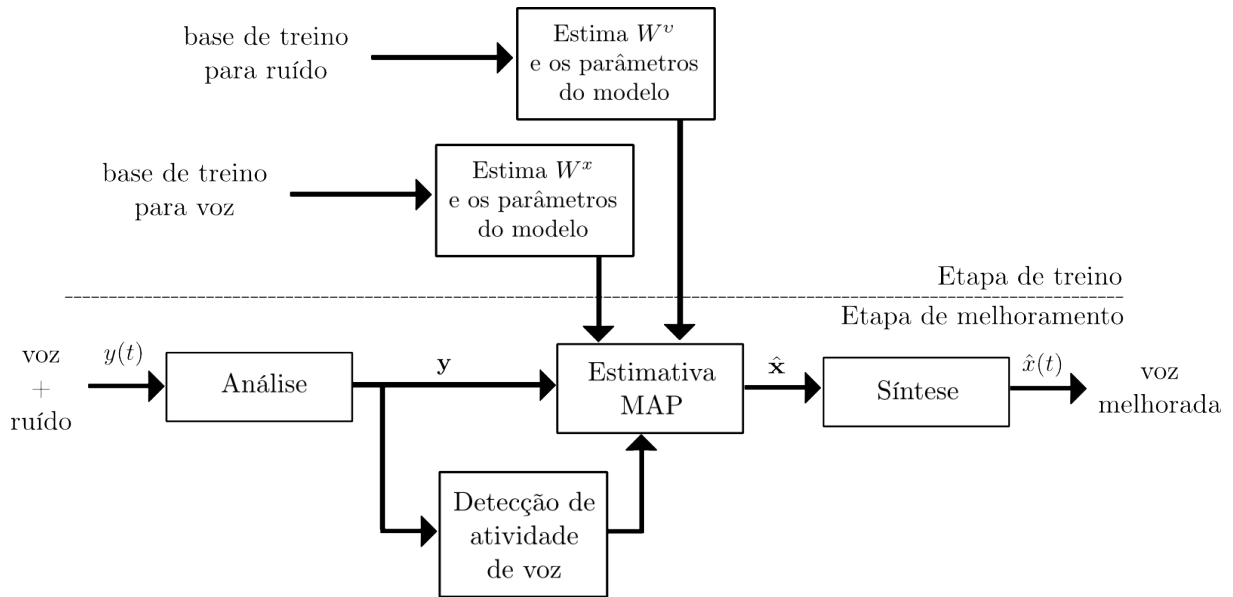


Figura 1 – Diagrama de blocos do algoritmo de melhoramento de voz ICA-MAP.

de probabilidade (pdf) é esparsa, dificultando a estimação pelo método do gradiente.

### 1.3 Objetivo

O objetivo principal deste trabalho é corrigir o problema de convergência do algoritmo ICA-MAP através de duas abordagens: a) limitando-se os valores do vetor gradiente e b) aproximando-se a derivada do modelo GGM em torno de zero (seu ponto crítico) por uma função contínua que permita a convergência pelo método do gradiente. Como objetivos específicos, serão realizados testes comparativos com outras técnicas clássicas, utilizando voz masculina e feminina, com diferentes potências de ruído, de modo a destacar as vantagens do método.

### 1.4 Publicação

Este trabalho foi publicado no XXXI Simpósio Brasileiro de Telecomunicações (SBrT). Seguem as informações:

CARMO, F. L. ; SALLES, EVANDRO O. T. . Uma Melhoria na Convergência do Método ICA-MAP para Remoção de Ruído em Sinal de Voz. In: Simpósio Brasileiro de Telecomunicações, 2013, Fortaleza. XXXI Simpósio Brasileiro de Telecomunicações (SBrT2013). Fortaleza, 2013. v. 1. p. 1-5. Palavras-chave: *speech enhancement*; ICA; *Non-gaussian noise*. Grande área: Engenharias / Área: Engenharia Elétrica / Subárea: Processamento Digital de Sinais. Setores de atividade: Pesquisa e desenvolvimento científico. Referências adicionais: Classificação do evento: Nacional; Brasil/ Português; Meio de



divulgação: Vários; Homepage: <[http://www.sbrt.org.br/sbrt2013/index.php?option=com\\_content&view=article&id=69&Itemid=66&lang=br](http://www.sbrt.org.br/sbrt2013/index.php?option=com_content&view=article&id=69&Itemid=66&lang=br)>; Série: 1. Esse artigo apresenta uma modificação feita no algoritmo ICA-MAP a fim de melhorar sua convergência. Foi observado, através de testes, que é possível limitar a magnitude do vetor gradiente, usado para estimar os parâmetros do modelo de denoising, e assim melhorar a estabilidade do algoritmo. Tal adaptação pode ser entendida como uma restrição no problema de otimização original (...). DOI: <<http://dx.doi.org/10.14209/sbrt.2013.218>>.

## 1.5 Divisão da dissertação

Este trabalho é organizado como segue. O Capítulo 2 introduz os conceitos básicos sobre análise de componentes principais (PCA) e ICA linear. Mostra como PCA e branqueamento podem ser usados como pré-processamento da ICA. Alguns conceitos sobre estatísticas de alta ordem são também introduzidos, a fim de mostrar um algoritmo simples e rápido baseado em iterações de ponto fixo, conhecido como fastICA, para estimar o modelo ICA linear.

O Capítulo 3 apresenta o algoritmo ICA-MAP e como é estimado. Descreve a estimativa MAP baseado em ICA linear e a solução iterativa com gradiente ascendente. A modelagem dos sinais de voz e ruído é realizada com GGM, e é mostrado como realizar a estimação dos parâmetros  $\beta$ , que indica o grau de não gaussianidade da distribuição. Na Seção 3.3 são propostas duas soluções para o problema de convergência do algoritmo ICA-MAP devido à característica esparsa dos dados.

O Capítulo 4 explicita como os experimentos foram realizados, bem como os resultados experimentais. Os algoritmos ICA-MAP com as modificações propostas são comparados com outros algoritmos de melhoramento de voz. Por fim, algumas observações finais são apresentadas no Capítulo 5

## 2 Análise de Componentes Independentes

### 2.1 Introdução

Análise de componentes independentes (ICA) é uma técnica cujo objetivo é expressar um conjunto de variáveis aleatórias como uma combinação de um outro conjunto de variáveis aleatórias tão independentes quanto possível. Tal combinação pode ser linear, que é a abordagem clássica da ICA, ou não-linear. Duas aplicações mais comuns de ICA são a separação cega de fontes e a extração de características.

### 2.2 Análise de componentes independentes

Originalmente, a ICA foi desenvolvida para tratar do problema do *cocktail-party*, que é discutido em (YOST; POPPER; FAY, 2008). Imagine-se, por exemplo, uma sala com 3 pessoas falando simultaneamente, e 3 microfones fixados em diferentes lugares desta sala, como ilustra a Figura 2. Cada microfone captura um sinal no tempo, denotados por

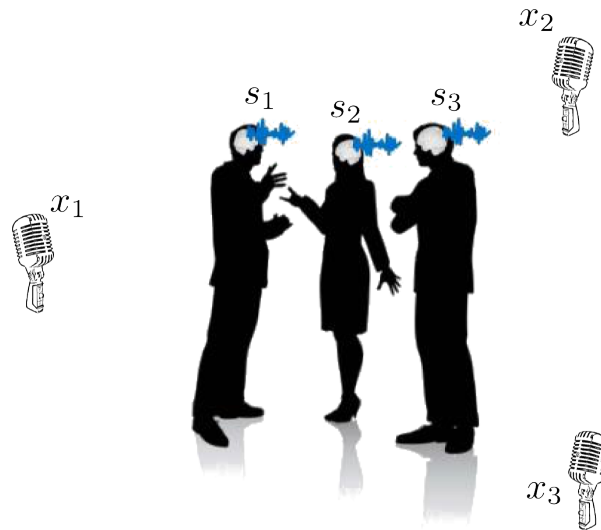


Figura 2 – Ilustração do problema *cocktail-party*.

$x_1(t)$ ,  $x_2(t)$  e  $x_3(t)$ . Cada um destes sinais é uma soma ponderada das vozes emitidas pelos 3 locutores, denotadas como  $s_1(t)$ ,  $s_2(t)$  e  $s_3(t)$ . Supondo linearidade, pode-se expressar essa situação como

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \\ x_3(t) &= a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t), \end{aligned} \tag{2.1}$$

onde  $a_{ij}$  são parâmetros que dependem das distâncias dos microfones para os locutores. O objetivo é estimar os sinais de voz originais  $s_j(t)$  usando apenas os sinais gravados  $x_i(t)$ . Como não são conhecidos nem os parâmetros  $a_{ij}$ , nem as fontes  $s_j$ , o problema é consideravelmente difícil. É interessante salientar que, por exemplo, se alguém está em uma sala com muitas pessoas falando simultaneamente e com música, essa pessoa é capaz de distinguir cada uma das vozes e sons, e é exatamente essa tarefa que a ICA tenta desempenhar. Uma possível abordagem é utilizar as propriedades estatísticas dos sinais  $s_j(t)$  para estimar os parâmetros  $a_{ij}$ .

Muitas vezes, é suficiente assumir que os sinais  $s_j(t)$  são estatisticamente independentes em cada instante de tempo, apesar de, em alguns casos, isto não ser verdade na prática, mas também não é uma suposição tão irrealista como parece. A Figura 3 mostra uma estimativa das componentes independentes usando apenas as misturas e a suposição que as fontes são independentes.

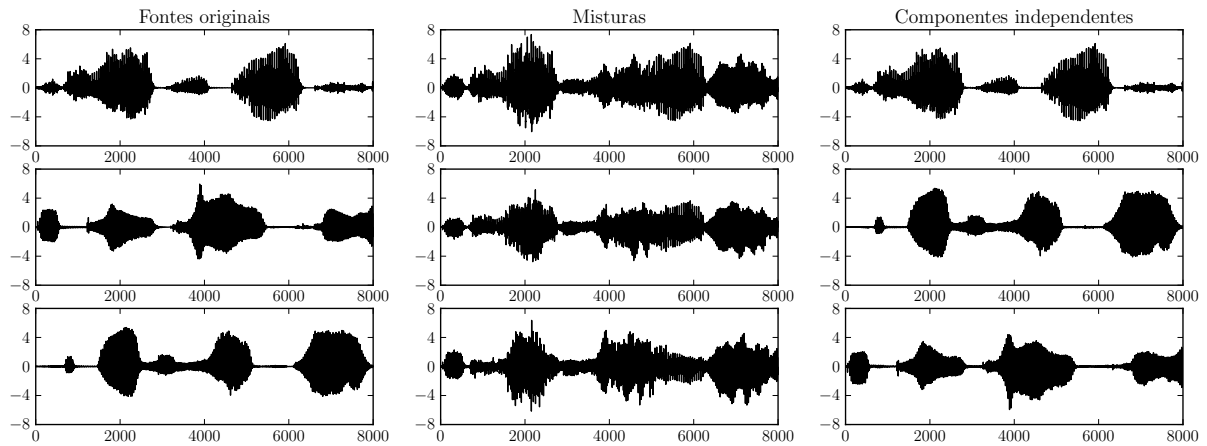


Figura 3 – Exemplo do *cocktail-party*. As fontes originais  $s_j$ , as misturas  $x_i$  e as componentes independentes estimadas.

Outra aplicação para ICA é a extração de características, que consiste em encontrar uma representação adequada para áudio, imagem, ou outro tipo de dados, a fim de realizar tarefas como compressão ou remoção de ruído (*denoising*), por exemplo.

## 2.3 Análise das Componentes Principais e Branqueamento

Análise das componentes principais, ou PCA, é uma técnica clássica utilizada em análise de dados estatísticos, extração de características e compressão de dados. PCA está intimamente relacionada com a transformada de Karhunen-Loève (KLT) (FUKUNAGA, 1990). A KLT resulta em uma representação linear de dados aleatórios como uma série convergente de funções ortogonais baseadas na função de covariância. Diferentemente da KLT, a técnica PCA trabalha com redução na dimensão dos dados. Entretanto, deve-se

primeiramente introduzir os conceitos de diagonalização e branqueamento da matriz de variância/covariância de uma variável aleatória.

As componentes de um vetor  $n$ -dimensional  $\mathbf{z}$  são mutuamente descorrelacionadas se

$$C_z = E\{(\mathbf{z} - \mathbf{m}_z)(\mathbf{z} - \mathbf{m}_z)^T\} = D \quad (2.2)$$

onde  $E\{\cdot\}$  é o operador valor esperado,  $C_z$  é a matriz de covariância,  $\mathbf{m}_z$  é o vetor de médias de  $\mathbf{z}$  e  $D$  é uma matriz diagonal com as variâncias de  $\mathbf{z}$ , ou seja,  $D = \text{diag}(\sigma_{z_1}^2, \sigma_{z_2}^2, \dots, \sigma_{z_n}^2)$ . Em particular, se o vetor tem média zero e matriz de variância/covariância unitária, possivelmente multiplicada por uma variância constante  $\sigma^2$ , ele é dito ser branqueado ou esférico. Ou seja, um vetor branqueado deve atender às condições

$$\mathbf{m}_z = \mathbf{0}, \quad C_z = \mathbf{I}, \quad (2.3)$$

onde  $\mathbf{I}$  é a matriz identidade. A Figura 4 ilustra um processo de branqueamento, onde  $x_1$  e  $x_2$  formam o sistema de coordenadas original,  $\Phi_1$  e  $\Phi_2$  o sistema de coordenadas dos dados branqueados e  $\mathbf{M}$  é o vetor de médias.

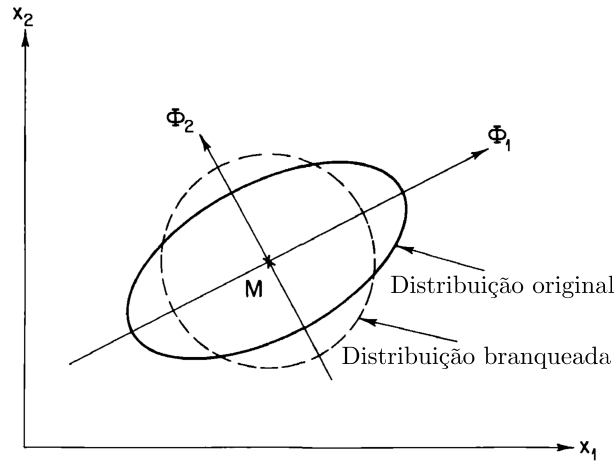


Figura 4 – Branqueamento. Adaptado de (FUKUNAGA, 1990)

Com isso em mente, pode-se introduzir PCA. Dado um conjunto de medidas multidimensionais, o objetivo é encontrar o menor conjunto de variáveis, com menos redundância, que dá uma representação tão boa quanto possível.

Em termos matemáticos, considera-se a combinação linear

$$y_1 = \sum_{k=1}^n w_{k1} x_k = \mathbf{w}_1^T \mathbf{x}, \quad (2.4)$$

onde  $\mathbf{w}_1$  é um vetor de coeficientes escalares e  $\mathbf{x}$  é um vetor aleatório com  $n$  elementos. A variável  $y_1$  é chamada de componente principal de  $\mathbf{x}$  se a variância de  $y_1$  é máxima. Para

evitar que o vetor de pesos  $\mathbf{w}_1$  cresça indefinidamente, limita-se sua norma euclidiana, ou seja,  $\|\mathbf{w}_1\| = 1$ . Com isso, o critério PCA a ser maximizado é

$$\begin{aligned} J_1^{PCA}(\mathbf{w}_1) &= E\{y_1^2\} = E\{(\mathbf{w}_1^T \mathbf{x})^2\} = \mathbf{w}_1^T E\{\mathbf{x}\mathbf{x}^T\} \mathbf{w}_1 = \mathbf{w}_1^T C_{\mathbf{x}} \mathbf{w}_1, \\ &\text{tal que } \|\mathbf{w}_1\| = 1, \end{aligned} \quad (2.5)$$

onde o vetor  $\mathbf{x}$  é primeiramente centrado, subtraindo sua média, ou seja, fazendo  $\mathbf{x} \leftarrow \mathbf{x} - E\{\mathbf{x}\}$ , e a estimação da média  $E\{\mathbf{x}\}$  é obtida através das amostras disponíveis.

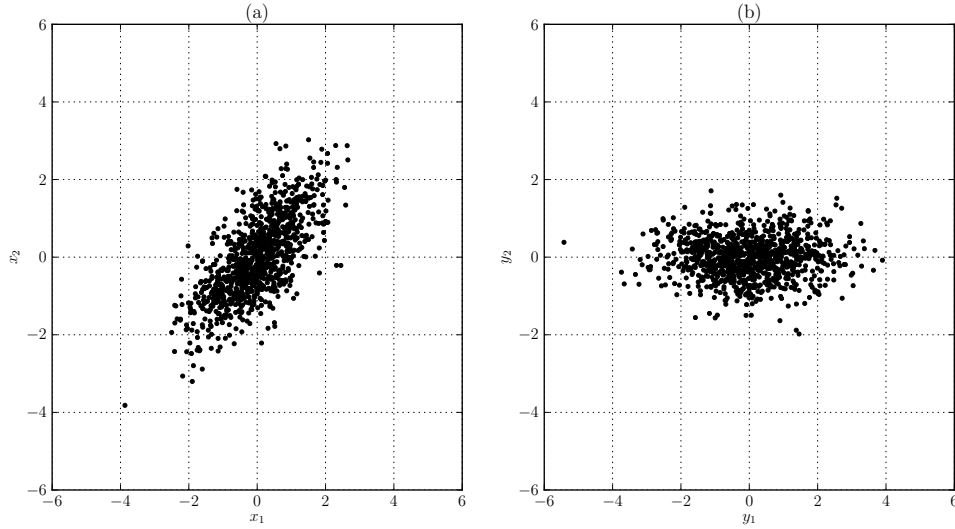


Figura 5 – Exemplo de PCA bidimensional. (a) Dados originais  $\mathbf{x}$ , e (b) as componentes principais  $\mathbf{y}$ .

É conhecido da álgebra linear (FUKUNAGA, 1990) que a solução do problema PCA é dada em termos dos autovetores  $\mathbf{e}_1, \dots, \mathbf{e}_n$  da matriz de covariância  $C_{\mathbf{x}}$ . E a ordem dos autovetores é dada pelos autovalores correspondentes  $d_1, \dots, d_n$ , tal que satisfaça  $d_1 \geq d_2 \geq \dots \geq d_n$ . Então, a solução de (2.5) é

$$\mathbf{w}_1 = \mathbf{e}_1, \quad (2.6)$$

e a componente principal é dada por  $y_1 = \mathbf{e}_1^T \mathbf{x}$ .

Como PCA trabalha com redução de dimensão, é essencial que  $\mathbf{x}$  seja mutuamente correlacionado, para que haja redundância, pois se as componentes são decorrelacionadas, nada pode ser feito por PCA. Em conexão com ICA, pode-se usar PCA como um pré-processamento. Dessa forma, o algoritmo ICA pode ser simplificado, já que PCA maximiza as estatísticas de segunda ordem dos dados. Assim, o ICA trata apenas das estatísticas de mais alta ordem, que serão apresentadas na Seção 2.4.

Comparando as Figuras 4 e 5, pode-se observar que o branqueamento é essencialmente decorrelação (PCA) seguida por um escalamento. Logo, a transformação de branqueamento pode ser realizada através de uma transformação linear semelhante à PCA.

O problema então é encontrar uma matriz  $V$  que transforma um vetor  $\mathbf{x}$  em outro vetor  $\mathbf{y}$  tal que  $\mathbf{y} = V\mathbf{x}$  seja branco. Tome-se  $E$  como sendo a matriz cujas colunas são os autovetores da matriz de covariância  $C_{\mathbf{x}}$ , que pode ser computada a partir das amostras de  $\mathbf{x}$ . Tome-se também a matriz  $D = \text{diag}(d_1, d_2, \dots, d_n)$ , que é a matriz cuja diagonal é formada pelos autovalores correspondentes. Então a transformação de branqueamento torna-se

$$V = D^{-\frac{1}{2}} E^T, \quad (2.7)$$

onde  $(\cdot)^T$  é o operador transposição. No caso do branqueamento não existe redução de dimensão dos dados.

## 2.4 Estatísticas de alta ordem

A análise de componentes independentes requer o uso de estatísticas de alta ordem (HYVÄRINEN; KARHUNEN; OJA, 2004). Seu uso pode ser direto, através de momentos estatísticos e cumulantes, ou indireto, através de não-linearidades, como por exemplo numa expansão em série de Taylor de uma função não-linear.

Considere-se uma variável aleatória  $x$  com função densidade de probabilidade (pdf)  $p_x(x)$ . O  $j$ -ésimo momento  $\alpha_j$  de  $x$  é definido como

$$\alpha_j = E\{x^j\} = \int_{-\infty}^{\infty} \xi^j p_x(\xi) d\xi, \quad j = 1, 2, \dots, \quad (2.8)$$

e o  $j$ -ésimo momento central  $\mu_j$  de  $x$  é definido como

$$\mu_j = E\{(x - \alpha_1)^j\} = \int_{-\infty}^{\infty} (\xi - m_x)^j p_x(\xi) d\xi, \quad j = 1, 2, \dots, \quad (2.9)$$

onde  $m_x$  é a média de  $x$ , que é igual a  $\alpha_1$ . O segundo momento  $\alpha_2 = E\{x^2\}$  é a potência média de  $x$ . Os momentos centrais  $\mu_0 = 1$  e  $\mu_1 = 0$  não têm significado, mas  $\mu_2 = \sigma^2$  é a variância de  $x$ . O momento central de terceira ordem  $\mu_3 = E\{(x - m_x)^3\}$  é conhecido como *skewness* e é útil para medir a assimetria da pdf. Para pdf's simétricas, a *skewness* é igual a zero.

O momento de quarta ordem é aplicado em alguns algoritmos ICA devido à sua simplicidade. Momentos estatísticos maiores que o de quarta ordem são raramente usados. Ao invés de utilizar o momento central  $\mu_4 = E\{(x - m_x)^4\}$ , uma outra estatística de quarta ordem é usualmente empregada, conhecida como *kurtosis*. A *kurtosis* é obtida através da teoria dos cumulantes, que será apresentada a seguir, e é definida, no caso de média zero, por

$$\text{kurt}(x) = E\{x^4\} - 3(E\{x^2\})^2, \quad (2.10)$$

ou, alternativamente, a *kurtosis* normalizada

$$\tilde{\kappa}(x) = \frac{E\{x^4\}}{(E\{x^2\})^2} - 3. \quad (2.11)$$

Para dados branqueados, tem-se que

$$\text{kurt}(x) = \tilde{\kappa}(x) = E\{x^4\} - 3. \quad (2.12)$$

Existem algumas propriedades da *kurtosis* que são muito úteis. São elas

$$\text{kurt}(x + y) = \text{kurt}(x) + \text{kurt}(y) \quad (2.13)$$

e

$$\text{kurt}(cx) = c^4 \text{kurt}(x), \quad (2.14)$$

onde  $c$  é um parâmetro escalar. Uma característica importante da *kurtosis* é que ela é a quantidade estatística mais simples para indicar a não-gaussianidade de uma variável aleatória. Em termos de *kurtosis* normalizada, para uma variável aleatória gaussiana, a *kurtosis* é zero, que na literatura é chamada de mesokúrtica. No caso de *kurtosis* negativa, a densidade é mais plana, e é conhecido como subgaussiana ou platokúrtica. Já para *kurtosis* positivas, a densidade tem um pico em zero, e é conhecida como supergaussiana ou leptokúrtica.

Agora, a fim de definir cumulantes (HYVÄRINEN; KARHUNEN; OJA, 2004), assume-se que  $x$  é variável aleatória escalar real de média zero, contínua com pdf  $p_x(x)$ . A primeira função característica  $\psi(\omega)$  de  $x$  é definida como a transformada de *Fourier* de  $p_x(x)$

$$\psi(\omega) = E\{\exp(j\omega x)\} = \int_{-\infty}^{\infty} \exp(j\omega x) p_x(x) dx, \quad (2.15)$$

onde  $j = \sqrt{-1}$  e  $\omega$  é a variável transformada. Cada pdf é unicamente especificada pela sua função característica. Expandindo  $\psi(\omega)$  em série de Taylor,

$$\psi(\omega) = \int_{-\infty}^{\infty} \left( \sum_{k=0}^{\infty} \frac{x^k (j\omega)^k}{k!} \right) p_x(x) dx = \sum_{k=0}^{\infty} E\{x^k\} \frac{(j\omega)^k}{k!}. \quad (2.16)$$

Assim, os coeficientes desta expansão são os momentos de  $x$ , assumindo que eles existam. Por isso,  $\psi(\omega)$  é conhecido como função geradora de momentos. Mas, é desejável usar uma segunda função característica  $\phi(\omega)$  de  $x$ , que é definida como o logaritmo natural da primeira função característica, donde se obtém

$$\phi(\omega) = \ln[\psi(\omega)] = \ln[E\{\exp(j\omega x)\}]. \quad (2.17)$$

Os cumulantes  $\kappa_k$  de  $x$  são definidos de forma similar aos respectivos momentos como os coeficientes da expansão em série de Taylor da segunda função característica. Então o  $k$ -ésimo cumulante é

$$\kappa_k = (-j)^k \left. \frac{d^k \phi(\omega)}{d\omega^k} \right|_{\omega=0}. \quad (2.18)$$

Para uma variável aleatória de média zero, os primeiros quatro cumulantes são  $\kappa_1 = 0$  (média),  $\kappa_2 = E\{x^2\}$  (variância),  $\kappa_3 = E\{x^3\}$  (*skewness*) e  $\kappa_4 = E\{x^4\} - 3(E\{x^2\})^2$  (*kurtosis*).

## 2.5 Definição de ICA

Para definir ICA, rigorosamente, pode-se usar modelo estatístico de variáveis latentes (HYVÄRINEN; KARHUNEN; OJA, 2004). Observam-se  $n$  variáveis aleatórias  $x_1, \dots, x_n$ , que são modeladas como combinações lineares de  $n$  variáveis aleatórias  $s_1, \dots, s_n$ , então

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \dots + a_{in}s_n, \text{ para todo } i = 1, \dots, n, \quad (2.19)$$

onde  $a_{ij}$  são coeficientes reais, e, por definição, as componentes  $s_j$  são estatisticamente independentes mutuamente. Este é o modelo ICA básico, e descreve como os dados observados são gerados a partir de um processo de mistura das componentes  $s_j$ . As componentes  $s_j$  são latentes, pois não podem ser observadas diretamente. E os coeficientes de mistura  $a_{ij}$  são desconhecidos. Apenas as componentes  $x_i$  são conhecidas, e, portanto, deve-se estimar as componentes independentes  $s_j$  e os coeficientes  $a_{ij}$  usando somente  $x_i$ . E tal estimação deve ser feita com suposições as mais gerais possíveis. É mais conveniente usar notação matricial ao invés da utilizada em (2.19). Portanto, suponha-se  $\mathbf{x}$  como sendo o vetor aleatório cujos elementos são as misturas  $x_1, \dots, x_n$ . Da mesma forma, suponha-se  $\mathbf{s}$  como sendo o vetor aleatório com os elementos  $s_1, \dots, s_n$ , e  $A$  como sendo a matriz dos coeficientes de mistura  $a_{ij}$ . Usando a notação matricial, o modelo de mistura é escrito como

$$\mathbf{x} = A\mathbf{s} \quad (2.20)$$

ou

$$\mathbf{x} = \sum_{j=1}^n \mathbf{a}_j s_j, \quad (2.21)$$

onde  $\mathbf{a}_j$  é a  $j$ -ésima linha de  $A$ . A fim de estimar a matriz de mistura  $A$  e as fontes  $s_j$ , é necessário otimizar uma função custo baseada em uma medida de independência estatística das componentes independentes. As abordagens mais comuns são minimização da informação mútua, que é um conceito definido pela teoria da informação, e maximização da não-gaussianidade.

A definição acima é a mais básica possível. Tal modelo não leva em consideração o ruído, que em algumas aplicações é uma suposição mais realista. Esta suposição complica muito a estimação, e uma possível abordagem é remover o ruído antes de aplicar ICA. Outra possível alteração no modelo é considerar uma mistura não-linear, onde a dificuldade principal é que a solução não será única, a menos que se adicionem algumas restrições extra, como, por exemplo, o uso de uma regularização adequada. Pode-se utilizar o modelo *post-nonlinear*, que assume que os sinais observados são gerados por misturas lineares seguido por componentes não-lineares. Em outras aplicações, o número de misturas e componentes independentes não são iguais, mas para o presente caso, assume-se que são iguais.



### 2.5.1 Restrições do ICA

Para ter certeza que o modelo apresentado acima pode ser estimado, é necessário fazer algumas suposições e restrições.

1. As componentes são assumidas estatisticamente independentes em alta ordem estatística.

Esse é o princípio no qual o ICA se sustenta. E não é necessário muito mais que essa suposição para a técnica funcionar. O conceito de independência estatística entre variáveis aleatórias  $y_1, y_2, \dots, y_n$  é que a informação de uma variável  $y_i$  não dá nenhuma informação sobre  $y_j$ , para  $i \neq j$ . Suponha-se  $p(y_1, y_2, \dots, y_n)$  como a pdf conjunta de  $y_i$ , e  $p_i(y_i)$  a pdf marginal de  $y_i$  (MILLER; CHILDERS, 2004). Então,  $y_i$  é independente se, e somente se, a pdf é fatorável, ou seja

$$p(y_1, y_2, \dots, y_n) = p_1(y_1)p_2(y_2) \cdots p_n(y_n). \quad (2.22)$$

2. As componentes independentes devem ter distribuições não-gaussianas.

Sabe-se que os cumulantes de alta ordem da distribuição gaussiana são zero, e tal informação é essencial para a estimação do modelo ICA em (2.20). Logo, é impossível estimar as componentes independentes se as mesmas são gaussianas. Considerem-se duas componentes independentes gaussianas com pdf conjunta dada por

$$p(s_1, s_2) = \frac{1}{2\pi} \exp\left(-\frac{s_1^2 + s_2^2}{2}\right) = \frac{1}{2\pi} \exp\left(-\frac{\|\mathbf{s}\|^2}{2}\right), \quad (2.23)$$

e uma matriz de mistura  $A$  ortogonal. Assume-se que os dados foram branqueados. Aplicando a fórmula clássica de transformação linear de variáveis aleatórias (MILLER; CHILDERS, 2004) e sabendo que  $A^{-1} = A^T$  para matrizes ortogonais, obtém-se a densidade conjunta das misturas  $x_1$  e  $x_2$  como

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{\|A^T \mathbf{x}\|^2}{2}\right) |\det A^T|. \quad (2.24)$$

Como  $A$  é ortogonal, sabe-se que  $\|A^T \mathbf{x}\|^2 = \|\mathbf{x}\|^2$  e  $|\det A| = 1$ . Então

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{\|\mathbf{x}\|^2}{2}\right). \quad (2.25)$$

Ou seja, a matriz de mistura ortogonal não modifica a pdf quando a variável é gaussiana, e logo a matriz de mistura não pode ser estimada. Então o que acontece ao se estimar o modelo ICA quando algumas componentes são gaussianas e outras não-gaussianas? Neste caso, consegue-se estimar todas as componentes não-gaussianas, mas as componentes

gaussianas não podem ser separadas umas das outras, ou seja, são combinações lineares das componentes gaussianas. Com isso, é possível estimar o modelo ICA se apenas uma componente for gaussiana.

3. Por simplicidade, assume-se que a matriz de mistura desconhecida é quadrada.

Isso significa que o número de componentes independentes é igual ao número de misturas observadas. Isso simplifica muito a estimação, pois após estimar a matriz  $A$ , pode-se computar a inversa, dita  $B$ , e obter as componentes independentes fazendo-se  $\mathbf{s} = B\mathbf{x}$ , assumindo que a matriz de mistura é inversível. Se  $A$  não é inversível, significa que existem misturas redundantes, que podem ser omitidas, logo a matriz de mistura não pode ser quadrada. Dessa forma, o número de misturas não é igual ao número de componentes independentes.

### 2.5.2 Ambiguidades do ICA

No modelo ICA em (2.20) existem algumas ambiguidade ou indeterminações.

1. Não se pode determinar as variâncias (energias) das componentes independentes.

Tanto a matriz de mistura  $A$  quanto as componentes independentes  $\mathbf{s}$  são desconhecidas. Com isso, algum escalar que multiplica uma das fontes  $s_j$  pode ser cancelado dividindo a coluna correspondente  $\mathbf{a}_j$  de  $A$  pelo mesmo escalar. Pode-se representar esta situação como

$$\mathbf{x} = \sum_j \left( \frac{1}{\alpha_j} \mathbf{a}_j \right) (s_j \alpha_j). \quad (2.26)$$

onde  $\alpha_j$  é um escalar qualquer e corrige-se a magnitude das componentes independentes. Como cada componente é uma variável aleatória, a forma mais conveniente de corrigir é assumir cada componente com variância unitária  $E\{s_j^2\} = 1$ . Mas observe-se que isso não corrige o problema da ambiguidade do sinal. Pode-se ainda multiplicar uma componente independente por  $-1$ , sem afetar o modelo. Felizmente, essa ambiguidade é insignificante em muitas aplicações.

2. Não se pode determinar a ordem das componentes independentes.

Novamente, por  $A$  e  $\mathbf{s}$  serem desconhecidos, pode-se livremente mudar a ordem dos termos da soma em (2.21). Tal permutação pode ser modelada por uma matriz  $P$  e adicionada ao modelo, ficando

$$\mathbf{x} = AP^{-1}P\mathbf{s}, \quad (2.27)$$

onde  $P\mathbf{s}$  são as variáveis independentes originais, mas com outra ordem, e  $AP^{-1}$  é a nova matriz de mistura a ser estimada.

### 2.5.3 Ilustração do ICA

Para ilustrar o modelo ICA, considerem-se duas variáveis aleatórias independentes, que têm a seguinte distribuição de probabilidade

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}}, & \text{se } |s_i| \leq \sqrt{3} \\ 0, & \text{caso contrário.} \end{cases} \quad (2.28)$$

A densidade conjunta de  $s_1$  e  $s_2$  é uniforme, como mostra a Figura 6(a). Agora, procede-se a duas componentes independentes com a seguinte matriz de mistura

$$A_0 = \begin{pmatrix} 5 & 10 \\ 10 & 2 \end{pmatrix}. \quad (2.29)$$

Isto resulta em duas misturas  $x_1$  e  $x_2$ . Na Figura 6(b) observa-se a distribuição dessa mistura dentro de um paralelogramo, deixando as componentes de serem independentes. É possível prever o valor de  $x_2$ , por exemplo, a partir do valor de  $x_1$ . Tomando-se o valor máximo ou mínimo de  $x_2$ , o valor de  $x_1$  é totalmente determinado. No caso de  $s_1$  e  $s_2$ , ao fixar qualquer valor de  $s_2$ , a probabilidade de  $s_1$  é a mesma, ou seja, são independentes.

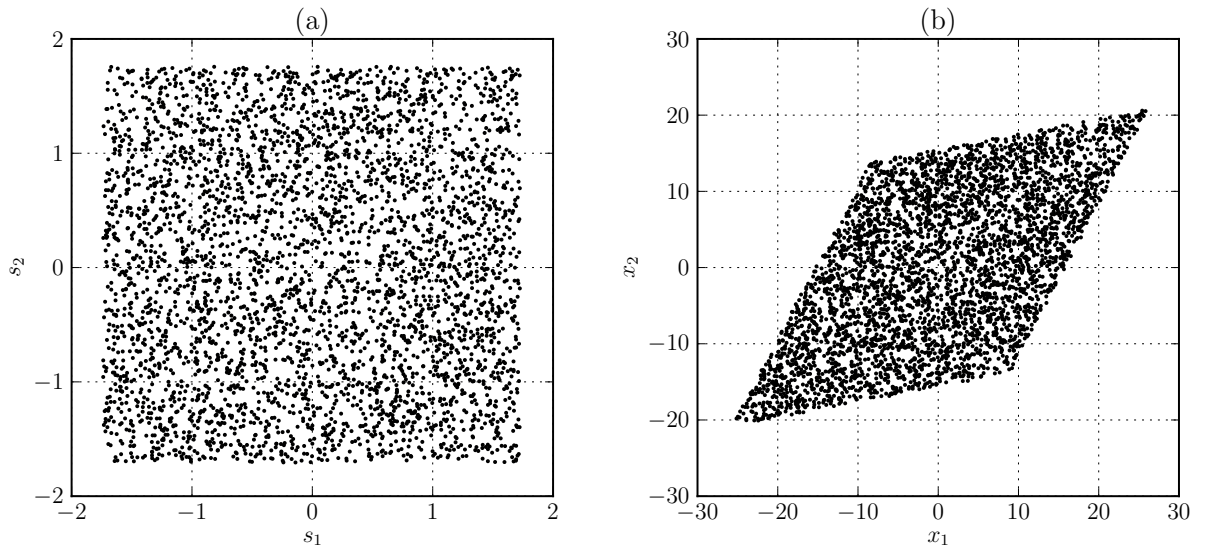


Figura 6 – Exemplo de ICA.

O problema de estimação do modelo ICA é agora encontrar a matriz de mistura  $A$  usando apenas a informação contida nas misturas  $x_1$  e  $x_2$ . Neste caso, pode-se ver que o problema pode ter uma solução intuitiva, utilizando as direções das bordas do paralelogramo da Figura 6(b), que estão nas colunas de  $A_0$  em (2.29). A princípio, pode-se estimar a densidade conjunta de  $x_1$  e  $x_2$ , e então localizar as bordas. Entretanto, esse exemplo foi dado apenas para fins didáticos, pois nem sempre é possível encontrar as bordas da densidade conjunta. Portanto, é necessário um método que funcione para qualquer distribuição das componentes independentes, e funcione de forma rápida e confiável.

## 2.6 Algoritmo fastICA

Muitas das soluções sugeridas para o modelo ICA usam o cumulante de quarta ordem, ou *kurtosis* dos sinais. Em (HYVÄRINEN; OJA, 1997), é introduzido um algoritmo simples e eficiente, baseado em um esquema de iterações de ponto fixo que encontra o extremo local da *kurtosis* de uma combinação linear das variáveis observáveis. É conhecido de (DELFOSSSE; LOUBATON, 1995) que encontrar o extremo local da *kurtosis* é equivalente a estimar as componentes independentes não-gaussianas.

O objetivo é encontrar uma combinação linear para a observação  $x_i = \mathbf{w}^T \mathbf{x}$ , de modo que se atinja a *kurtosis* mínima ou máxima, onde  $\mathbf{w}$  é um vetor de pesos e  $x_i$  é branqueado. Isso tem sentido apenas se a norma de  $\mathbf{w}$  é de alguma forma limitada. Então assume-se  $\|\mathbf{w}\| = 1$ . Usando a matriz de mistura  $A$  ortogonal, define-se  $\mathbf{z} = A^T \mathbf{w}$ . Então  $\|\mathbf{z}\| = 1$ . Usando (2.20) e as propriedades da *kurtosis* em (2.13) e (2.14), tem-se

$$\text{kurt}(\mathbf{w}^T \mathbf{x}) = \text{kurt}(\mathbf{w}^T A \mathbf{s}) = \text{kurt}(\mathbf{z}^T \mathbf{s}) = \sum_{i=1}^n z_i^4 \text{kurt}(s_i), \quad (2.30)$$

com a restrição  $\|\mathbf{w}\| = \|\mathbf{z}\| = 1$ , onde  $n$  é o número de componentes. Por simplicidade, assume-se, por enquanto, que a mistura contém ao menos uma componente cuja *kurtosis* seja positiva, e ao menos uma cuja *kurtosis* seja negativa. Então, como é mostrado em (DELFOSSSE; LOUBATON, 1995), os extremos em (2.30) são os vetores da base canônica  $\mathbf{z} = \pm \mathbf{e}_j$ , isto é, vetores cujas componentes são todas zero, exceto uma, que é igual a  $\pm 1$ . Com isso,  $\mathbf{w} = A\mathbf{z} = A\mathbf{e}_j = \mathbf{a}_j$ , que é uma coluna da matriz de mistura ortogonal  $A$ . Ou seja, encontrar os vetores de pesos  $\mathbf{w}$  para cada componente corresponde a estimar cada coluna da matriz de mistura  $A$ . E, através disso, obtém-se  $\mathbf{w}^T \mathbf{x} = (\mathbf{a}_j)^T \mathbf{x} = s_j$ . A equação (2.30) também mostra que não é possível estimar componentes gaussianas com essa abordagem, pois a *kurtosis* para elas é zero.

A fim de estimar os extremos de  $\text{kurt}(\mathbf{w}^T \mathbf{x})$ , pode-se usar algum algoritmo baseado em redes neurais, como em (DELFOSSSE; LOUBATON, 1995) e (HYVARINEN; OJA, 1996). Porém, este tipo de algoritmo converge lentamente e depende de uma boa escolha da taxa de aprendizado. Portanto, algoritmos de iteração de ponto fixo são uma alternativa, pois o aprendizado é mais rápido e mais confiável.

A função contraste pode ser escrita usando (2.10) e (2.30), e levando em conta que os dados estão branqueados, temos

$$\text{kurt}(\mathbf{w}^T \mathbf{x}) = E\{(\mathbf{w}^T \mathbf{x})^4\} - 3(E\{(\mathbf{w}^T \mathbf{x})^2\})^2 = E\{(\mathbf{w}^T \mathbf{x})^4\} - 3\|\mathbf{w}\|^4. \quad (2.31)$$

A restrição  $\|\mathbf{w}\| = 1$  deve ser considerada utilizando uma função penalidade (HYVARINEN; OJA, 1996). Então a função contraste final é

$$J(\mathbf{w}) = E\{(\mathbf{w}^T \mathbf{x})^4\} - 3\|\mathbf{w}\|^4 + F(\|\mathbf{w}\|^2), \quad (2.32)$$

onde  $F$  é a função de regularização, devido à restrição. Com isso, como é mostrado em (HYVÄRINEN; OJA, 1997), a regra de aprendizado do ponto fixo  $\mathbf{w}$  fica sendo

$$E\{\mathbf{x}(\mathbf{w}^T \mathbf{x})^3\} - 3\|\mathbf{w}\|^2 \mathbf{w} + f(\|\mathbf{w}\|^2) \mathbf{w} = 0, \quad (2.33)$$

que também pode ser escrita como

$$\mathbf{w} = \text{escalar} \times (E\{\mathbf{x}(\mathbf{w}^T \mathbf{x})^3\} - 3\|\mathbf{w}\|^2 \mathbf{w}), \quad (2.34)$$

onde *escalar* não é significativo, pois o que importa é a direção do lado direito de (2.34), e não a norma de  $\mathbf{w}$ . Então, com (2.34) é possível desenvolver o algoritmo ICA de ponto fixo.

### 2.6.1 O algoritmo de ponto fixo fastICA

Assumindo um conjunto de amostras em um vetor aleatório branqueado  $\mathbf{x}$ , deve-se seguir o seguinte algoritmo:

1. Escolher um vetor aleatório inicial  $\mathbf{w}(0)$  de norma 1. Fazer  $k = 1$ , onde  $k$  é o número de iterações.
2. Fazer  $\mathbf{w}(k) = E\{\mathbf{x}(\mathbf{w}(k-1)^T \mathbf{x})^3\} - 3\mathbf{w}(k-1)$ . A expectância pode ser estimada usando um número grande de amostras do vetor  $\mathbf{x}$  (1000 pontos, por exemplo).
3. Fazer  $\mathbf{w}(k) = \mathbf{w}(k) - \bar{A}\bar{A}^T \mathbf{w}(k)$ , onde  $\bar{A}$  é uma matriz cujas colunas são as colunas previamente encontradas da matriz de mistura  $A$ . Dividir  $\mathbf{w}(k)$  por sua norma.
4. Se  $|\mathbf{w}(k)^T \mathbf{w}(k-1)|$  não está suficientemente próximo de 1, fazer  $k = k + 1$  e voltar ao passo 2. Caso contrário, a saída é o vetor  $\mathbf{w}$ . Acrescenta-se  $\mathbf{w}$  à  $\bar{A}$  como uma coluna. Se todas as componentes independentes foram encontradas, finalizar. Caso contrário, retornar ao passo 1.

Como a matriz de mistura deve ser ortogonal, o passo 3 realiza uma projeção de ortogonalização. Entretanto, quando se está estimando a primeira componente independente, o processo de ortogonalização não é necessário, pois  $\mathbf{w}(k)$  não será alterada pela projeção.

O algoritmo acima estima o modelo ICA por uma abordagem de maximização de não-gaussianidade, e faz isso utilizando o cumulante de quarta ordem  $\text{kurt}(v) = E\{v^4\} - 3(E\{v^2\})^2$  como função contraste. Entretanto, é mostrado em (HYVARINEN, 1997) que a função contraste pode ser derivada de qualquer função não quadrática. Na prática, isso é uma vantagem, pois a *kurtosis* cresce muito rápido, e, portanto, é muito sensível a *outliers* (observações muito afastadas das demais). Para uma função não quadrática suficientemente regular  $G$ , a função contraste é

$$J_G(x) = E_x\{G(x)\} - E_v\{G(\sigma_x v)\}, \quad (2.35)$$

onde  $v$  é uma variável gaussiana com variância unitária,  $\sigma_x = \sqrt{E\{x^2\}}$  e assim  $\sigma_x v$  é uma variável gaussiana com mesma variância de  $x$ . Observa-se que (2.35) pode ser considerado uma generalização da *kurtosis*, pois fazendo  $G(x) = x^4$ , volta-se ao caso em (2.31). Porém, ao escolher  $G(x) = \tanh(u)$ , por exemplo, os valores afastados de zero não são grandes, logo são robustos a *outliers*.

Outra medida de não-gaussianidade que pode ser utilizada para estimar o modelo ICA é a negentropia, que vem da teoria da informação e pode ser entendida como a distância da gaussianidade. Assumindo uma variável multidimensional  $\mathbf{y}$ , a negentropia é definida como

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}), \quad (2.36)$$

onde

$$H(\mathbf{y}) = - \int p_{\mathbf{y}}(\xi) \log p_{\mathbf{y}}(\xi) d\xi \quad (2.37)$$

é a entropia diferencial de uma densidade  $p_{\mathbf{y}}$ . Sabe-se que a variável gaussiana tem a maior entropia dentre todas variáveis de mesma variância, por isso a negentropia pode ser entendida como medida de não-gaussianidade. Entretanto, na prática, é necessário aproximar a negentropia. O método clássico é aproximar pelos cumulantes de alta ordem. Porém, o problema com os *outliers* retorna. A abordagem mais conveniente é aproximar por funções não quadráticas e não-polinomiais, como em (2.35).

Uma outra possível abordagem é a utilização da informação mútua, um outro conceito da teoria da informação que é uma medida de dependência. Então, ao invés de assumir independência estatística entre os dados, define-se o modelo ICA como uma decomposição linear que minimiza a dependência entre as variáveis do vetor. Isso não será detalhado, pois, para o caso atual, é suficiente a abordagem por maximização da não-gaussianidade.

## 3 ICA-MAP

### 3.1 Introdução

O algoritmo de melhoramento de voz ICA-MAP foi proposto em (ZOU *et al.*, 2008). O método consiste em retirar o ruído utilizando duas transformações ICA lineares, uma para o sinal de voz e outra para o ruído, a partir de uma estimativa MAP (*Maximum a posteriori*).

### 3.2 A abordagem ICA-MAP

#### 3.2.1 Estimativa MAP

Considerando variáveis aleatórias escalares,  $x$  representa o sinal limpo e  $v$  a parcela de ruído aditivo. Assume-se que o sinal observado  $y$  é uma versão ruidosa de  $x$ , ou seja

$$y = x + v. \quad (3.1)$$

O interesse é estimar o sinal limpo  $x$  usando apenas o sinal observado  $y$ , por meio de uma função  $\hat{x} = g(y)$ . Para isso, utiliza-se o algoritmo de estimação MAP, que, diferentemente da estimação ML (*Maximum Likelihood*), leva em consideração a probabilidade do sinal corrompido  $y$  condicionado a uma realização do sinal limpo  $x$  e a função densidade de probabilidade *a priori* do sinal limpo. Assim, utilizando o logaritmo da densidade de probabilidade, o algoritmo resulta em

$$\hat{x} \leftarrow \arg \max_x (\ln p_y(y | x) + \ln p_x(x)), \quad (3.2)$$

onde o termo  $p_y(y | x)$  é a distribuição condicional de  $y$  dado  $x$ , que é a probabilidade do ruído avaliada em  $y - x$ , ou seja,  $p_y(y | x) = p_v(y - x)$ , e  $p_x(x)$  é a distribuição *a priori* de  $x$ . A fim de aplicar a transformação ICA, os sinais de áudio devem ser segmentados em *frames*, cada um composto por  $N$  amostras, onde o valor de  $N$  deve garantir a estacionariedade do sinal dentro do *frame*. Agora, é possível escrever, para o sinal ruidoso  $y$ , o seguinte vetor  $N$ -dimensional de observações,

$$\mathbf{Y} = \{\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(M-1), \mathbf{y}(M)\},$$

onde  $\mathbf{y}(t) = [y(t), y(t-1), \dots, y(t-N+1)]^T$  e  $(\cdot)^T$  significa o operador de transposição de matriz. O vetor de observações  $\mathbf{Y}$  é uma matriz de dimensão  $N \times M$ . Os *frames* são sobrepostos, avançando uma amostra a cada *frame*, para capturar melhor as características

do sinal. Usando a mesma notação para o sinal limpo, a estimativa do sinal limpo  $\hat{\mathbf{x}}$ , pela lei dos grandes números ( $M \rightarrow \infty$ ), e de acordo com (3.2), é

$$\begin{aligned} \{\hat{\mathbf{x}}(1), \dots, \hat{\mathbf{x}}(M)\} &\leftarrow \arg \max_{\hat{\mathbf{x}}} \left( \sum_t \ln p_v(\mathbf{y}(t) - \hat{\mathbf{x}}(t)) + \sum_t \ln p_x(\hat{\mathbf{x}}(t)) \right) \\ &\Leftrightarrow \arg \max_{\hat{\mathbf{x}}} (E\{\ln p_v(\mathbf{y}(t) - \hat{\mathbf{x}}(t))\} + E\{\ln p_x(\hat{\mathbf{x}}(t))\}). \end{aligned} \quad (3.3)$$

Daqui para frente será omitido o índice  $t$  e se substituirá  $\hat{\mathbf{x}}$  por  $\mathbf{x}$ .

Em (ZOU; JANCOVIC; KOKUER, 2008) é mostrado que a maximização em (3.3) pode ser alcançada usando uma abordagem ICA linear, onde é possível aplicar transformações individuais para o sinal de voz e ruído. Através do algoritmo ICA, são obtidas as matrizes  $W^x$  e  $W^v$ , comumente chamadas matrizes de separação, que levam as componentes de voz e ruído para um espaço onde elas se tornam tão independentes quanto possível. E ao aplicar transformações individuais, a otimização é atingida individualmente para voz e ruído. Para realizar a estimação MAP utilizando essa transformação, usa-se a relação entre a função densidade de uma variável aleatória  $a$  e de uma versão transformada linearmente  $b = Wa$ , que é  $p_a(\mathbf{a}) = p_b(W\mathbf{a})|\det(W)|$ .

Considerando a estimativa de um único *frame*, o operador expectativa, que aparece em (3.3), pode ser removido e então o algoritmo MAP baseado em ICA resulta em

$$\begin{aligned} \mathbf{x} &\leftarrow \arg \max_{\mathbf{x}} (\ln p_{s_v}(W^v(\mathbf{y} - \mathbf{x})) + \ln p_{s_x}(W^x\mathbf{x}) + \\ &\quad + \ln |\det(W^v)| |\det(W^x)|), \end{aligned} \quad (3.4)$$

onde  $W^v$  e  $W^x$  são as matrizes de separação das componentes independentes do ruído e do sinal limpo, respectivamente. Já  $p_{s_v}$  e  $p_{s_x}$  são as pdf's das componentes independentes subjacentes do ruído  $v$  e do sinal limpo  $x$ , respectivamente. Considerando que as matrizes  $W^v$  e  $W^x$  são constantes para um determinado ruído e sinal limpo, o terceiro termo do lado direito em (3.4) pode ser omitido. A estimativa de um *frame* do sinal  $x(t)$  pode ser obtida usando o método do gradiente. É necessário lançar mão de um método iterativo pois não é possível alcançar uma fórmula fechada para a estimação. Então, derivando a expressão em (3.4) em relação a cada componente de  $\mathbf{x}$ , pode-se escrever a regra de estimação para a  $n$ -ésima amostra do *frame* como

$$x_n \leftarrow x_n + \lambda \sum_{i=0}^N [f'_v(W_{i\cdot}^v(\mathbf{y} - \mathbf{x}))w_{in}^v + f'_x(W_{i\cdot}^x\mathbf{x})w_{in}^x], \quad (3.5)$$

onde  $\lambda$  é o tamanho do passo,  $W_{i\cdot}$  é a  $i$ -ésima linha da matriz  $W$ ,  $w_{in} = W(i, n)$  e  $f'(s) = -\partial \ln p(s) / \partial s$ . O algoritmo mostrado na equação (3.5) é o caso geral do ICA-MAP e daqui em diante será chamado de ICA-MAP(gen).

### 3.2.2 Casos Especiais

Deve-se notar que tal algoritmo não faz nenhuma suposição sobre a natureza do ruído. Logo, ele pode ser utilizado para tratar de ruídos gaussianos e não-gaussianos.



Assumindo que o sinal de voz e o ruído possuem distribuição gaussiana, ou seja

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \quad (3.6)$$

onde  $\sigma^2$  é a variância, e sabendo que a variável aleatória gaussiana não é afetada por transformações lineares, a maximização em (3.3) pode ser realizada no espaço original, ou seja,  $W_x$  e  $W_v$  são iguais à matriz identidade. Considerando a estimativa de um único *frame*, tem-se, então

$$\mathbf{x} \leftarrow \arg \max_{\mathbf{x}} \ln p_v(\mathbf{y} - \mathbf{x}) + \ln p_x(\mathbf{x}). \quad (3.7)$$

Substituindo (3.6) em (3.4) e escrevendo a regra para cada componente do *frame*, é possível escrever

$$\begin{aligned} x_n &\leftarrow \arg \max_x (\ln p_v(y_n - x_n) + \ln p_x(x_n)) \\ &\Leftrightarrow \arg \min_x \left( \frac{1}{2\sigma_v^2} (y_n - x_n)^2 + \frac{1}{2\sigma_x^2} x_n^2 \right). \end{aligned} \quad (3.8)$$

A minimização equivale a resolver a equação anterior, ou seja, igualando a derivada a respeito de  $x_n$  a zero, ou seja

$$\frac{1}{\sigma_v^2} (x_n - y_n) + \frac{1}{\sigma_x^2} x_n = 0 \Leftrightarrow x_n = \frac{\sigma_x^2}{\sigma_v^2 + \sigma_x^2} y_n. \quad (3.9)$$

Tal estimativa é conhecida como filtragem de *Wiener*, e foi originalmente baseada no critério da minimização do erro quadrático médio (MMSE). Sabendo que o critério MMSE considera apenas os dois primeiros cumulantes (média e variância), e os outros cumulantes são iguais a zero para a distribuição gaussiana, ambos critérios, MAP e MMSE, convergem para a mesma regra de estimação.

Agora, considerando que apenas o ruído possui distribuição gaussiana, o termo  $E\{p_v(\mathbf{y} - \mathbf{x})\}$  em (3.3) não tem extremos. Então, a maximização dependerá apenas do termo  $E\{p_x(\mathbf{x})\}$ . E esse máximo é alcançado usando uma matriz de separação ICA de  $x$ , conforme (3.4). Como a distribuição gaussiana não se altera com transformações lineares, pode-se usar a mesma transformação do sinal limpo  $W_x$  no ruído gaussiano, ou seja, fazendo  $W_v = W_x = W$ . Então, fazendo tais modificações em (3.4), tem-se

$$\begin{aligned} \mathbf{x} &\leftarrow \arg \max_{\mathbf{x}} \left( \sum_{n=1}^N \ln p(W_n(\mathbf{y} - \mathbf{x})) + \ln p(W_n \mathbf{x}) + 2 \ln |det(W)| \right) \\ &\Leftrightarrow \sum_{n=1}^N \arg \max_{\mathbf{x}} (\ln p(W_n(\mathbf{y} - \mathbf{x})) + \ln p(W_n \mathbf{x})), \end{aligned} \quad (3.10)$$

onde  $W_{n.}$  é a  $n$ -ésima linha da matriz  $W$ .

Como há apenas um espaço de transformação, dado pela matriz  $W$ , pode-se fazer a estimação no espaço transformado, e então retornar ao espaço original. Para isso, assume-se

que  $W_n \mathbf{y} = z_n$  e  $W_n \mathbf{x} = s_n$ . Resolvendo-se para cada componente, temos

$$\begin{aligned} s_n &\leftarrow \arg \max_s (\ln p(z_n - s_n) + \ln p(s_n)) \\ &\Leftrightarrow \arg \min_s \left( \frac{1}{2\sigma^2} (z_n - s_n)^2 + f(s_n) \right) \end{aligned} \quad (3.11)$$

A minimização é equivalente a resolver a equação

$$\frac{1}{\sigma^2} (s_n - z_n) + f'(s_n) = 0. \quad (3.12)$$

Arranjando os termos, obtém-se

$$g^{-1}(s_n) = s_n + \sigma^2 f'(s_n), \quad (3.13)$$

onde  $s_n = g(z_n)$ . Portanto, basta inverter a função em (3.13) para encontrar a solução. Porém, a função a ser invertida envolve a densidade de  $s$ , que é uma variável aleatória não-gaussiana que será modelada por um modelo gaussiano generalizado, apresentado na seção 3.2.3; logo a função a ser invertida será não-linear. Neste caso, segundo (HYVÄRINEN, 1999), é possível obter uma aproximação de 1ª ordem (em relação ao ruído) do estimador

$$\hat{s}_n^* = z_n - \sigma^2 f'(z_n), \quad (3.14)$$

assumindo que  $f$  seja convexa e diferenciável. Observa-se que (3.14) deriva de (3.12) apenas substituindo o termo não observável  $f'(s_n)$  pela quantidade observável  $f'(z_n)$ , que são iguais na aproximação de 1ª ordem. O problema desta aproximação está na região próxima a zero, onde o sinal de  $\hat{s}_n^*$  é frequentemente diferente do sinal de  $y$ . Isso ocorre porque  $f'$  é, muitas vezes, descontínuo, ou até mesmo singular em 0. Isso dependerá dos parâmetros do modelo a ser usado. A fim de amenizar este problema, foi proposta a seguinte modificação (HYVÄRINEN, 1999)

$$\hat{s}_n^o = \text{sign}(z_n) \max(0, |y| - \sigma^2 |f'(z_n)|), \quad (3.15)$$

onde  $\text{sign}(z_n)$  significa o sinal de  $z_n$  e  $\max(a, b)$  significa o maior valor entre  $a$  e  $b$ . Portanto, temos o estimador exato de  $s$ , obtido pela inversão de (3.13), e suas aproximações em (3.14) e (3.15). A Figura 7 mostra os gráficos destas soluções, onde a solução exata foi estimada através de um algoritmo de ponto fixo, como mostrado em (COLLINS, 1990). A regra de estimação em (3.15) é conhecida como *Sparse Code Shrinkage*, e foi originalmente proposta e utilizada para imagem em (HYVÄRINEN, 1999), e posteriormente usada para melhoramento de voz em (LEE *et al.*, 2000) e (POTAMITIS; FAKOTAKIS; KOKKINAKIS, 2001), e utilizada com uma possível correlação do ruído em (HONG; ROSCA; BALAN, 2004).

Em (ZOU *et al.*, 2008), a capacidade da regra de estimação em (3.3) é avaliada. Em estimação bayesiana, a estimativa de um sinal desconhecido é realizada minimizando um risco condicional  $R$ , que é

$$R_x(\hat{x}) = E\{C(\hat{x}, x)\} = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} C(\hat{x}, x) p(x|y) dx \right] p(y) dy, \quad (3.16)$$

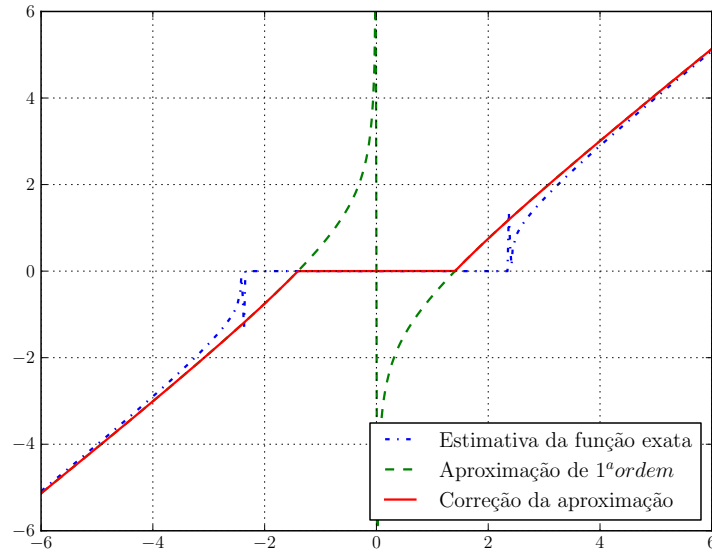


Figura 7 – Gráficos da solução exata (estimativa) e suas aproximações.

onde  $C(\hat{x}, x)$  é uma função custo de estimar o valor verdadeiro  $x$  como  $\hat{x}$ ,  $y$  é o sinal ruidoso observado e  $x$  é o valor verdadeiro escondido em  $y$ . Assume-se uma densidade *a priori*  $p(y)$  uniforme e  $C(\hat{x}, x)$  uniforme que penaliza quando a diferença  $|x - \hat{x}|$  é maior que um pequeno valor  $\epsilon/2$ . Com isso, o risco fica

$$R_x(\hat{x}) = E \left\{ \frac{1}{\epsilon} \frac{p(y|\hat{x})p(\hat{x})}{p(y)} \right\}. \quad (3.17)$$

Em (CARDOSO, 1997) e (ZOU *et al.*, 2008), temos que o máximo das expectâncias em (3.3) é igual ao valor negativo da entropia. A entropia pode ser aproximada por uma expansão de *Edgeworth* (CARDOSO, 1999) como uma soma da *kurtosis* e a entropia de uma variável gaussiana de mesma matriz de covariância. Com essas considerações, temos

$$E\{\ln p(\mathbf{y}|\mathbf{x})p(\mathbf{y})\}_{\max} \approx \left( \frac{1}{48} \kappa(\mathbf{s}_v)^2 + K_v \right) + \left( \frac{1}{48} \kappa(\mathbf{s}_x)^2 + K_x \right), \quad (3.18)$$

onde  $K_v$  e  $K_x$  são valores constantes, irrelevantes para a análise. Considerando (3.17), (3.18) e o fato que a *kurtosis* é zero para variáveis aleatórias gaussianas e a aumenta na medida que a distribuição se torna super-gaussiana, temos

$$R_{(x_{ng}, v_{ng})} < R_{(x_{ng}, v_g)} < R_{(x_g, v_g)}, \quad (3.19)$$

onde os índices *ng* e *g* denotam as distribuições não gaussianas e gaussianas, respectivamente. Como valores menores do risco significam melhor capacidade de melhoramento de voz, a Equação (3.19) mostra que a capacidade aumenta à medida que aumenta a não gaussianidade dos sinais de voz e ruído.

### 3.2.3 Modelo Gaussiano Generalizado

A *pdf* do algoritmo MAP em (3.4) é obtida através do modelo Gaussiano generalizado (GGM), usado em (LEE; LEWICKI, 2000). A escolha do modelo GGM é devido ao fato dele ser um bom ajuste para vários graus de não-gaussianidade, podendo modelar distribuições muito esparsas. A forma geral do modelo GGM é

$$p(x|m, \delta, \beta) = \frac{\omega(\beta)}{\delta} \exp \left[ -c(\beta) \left| \frac{x - m}{\delta} \right|^{\frac{2}{1+\beta}} \right], \quad (3.20)$$

onde

$$c(\beta) = \left[ \frac{\Gamma[3(1+\beta)/2]}{\Gamma[(1+\beta)/2]} \right]^{\frac{1}{1+\beta}}, \quad (3.21)$$

e

$$\omega(\beta) = \frac{\Gamma[3(1+\beta)/2]^{\frac{1}{2}}}{(1+\beta)\Gamma[(1+\beta)/2]^{\frac{3}{2}}}, \quad (3.22)$$

onde  $\Gamma[\cdot]$  é a função gama, e  $m$  e  $\delta$  denotam a média e o desvio padrão dos dados, respectivamente. O parâmetro  $\beta$  controla a não-gaussianidade da distribuição, está relacionado com a *kurtosis* e deve ser maior que  $-1$ . Esse modelo pode descrever distribuições gaussianas ( $\beta = 0$ ), sub-gaussianas ( $-1 < \beta < 0$ ) e super-gaussianas ( $\beta > 0$ ). Por exemplo, quando  $\beta = 1$ , a distribuição é Laplaciana, quando  $\beta \rightarrow -1$ , a distribuição tende para a uniforme, e para  $\beta \rightarrow \infty$  a distribuição tende para a função delta. A Figura 8 mostra alguns exemplos.

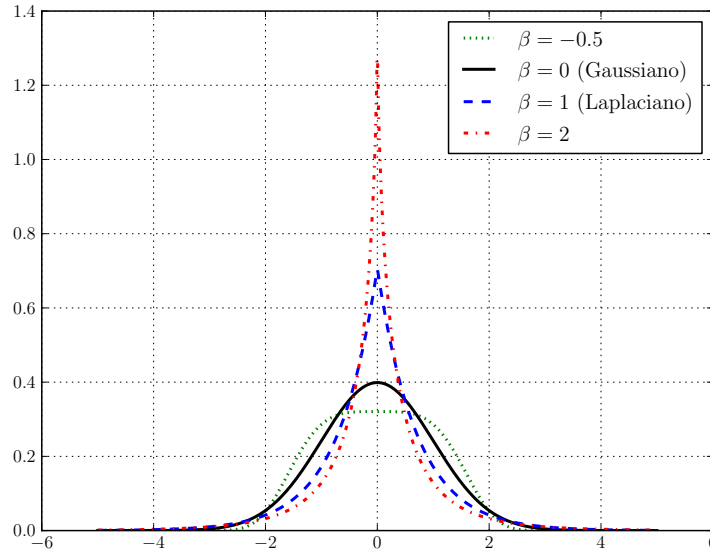


Figura 8 – Exemplos do modelo GGM com alguns valores de  $\beta$ .

Para média zero, as funções  $f'$  em (3.5), que são o diferencial do logaritmo das

*pdf*'s, podem ser expressas como

$$f'(x) = \frac{2c(\beta)}{\delta(1+\beta)} \left| \frac{x}{\delta} \right|^{\frac{1-\beta}{1+\beta}}. \quad (3.23)$$

A fim de estimar o parâmetro  $\beta$ , pode ser usado o método MAP. Utilizando as amostras do sinal de treinamento, tem-se

$$\beta \leftarrow \arg \max_{\beta} p(\beta|\mathbf{x}) \Leftrightarrow \arg \max_{\beta} p(\mathbf{x}|\beta)p(\beta), \quad (3.24)$$

onde

$$p(\mathbf{x}|\beta) = \prod_t \omega(\beta) \exp \left[ -c(\beta) |x(t)|^{\frac{2}{1+\beta}} \right] \quad (3.25)$$

e  $p(\beta)$  é a distribuição *a priori* de  $\beta$ . Como  $\beta > -1$ , é conveniente usar uma distribuição *gamma* deslocada para modelar  $p(\beta)$ , como se observa a seguir

$$\begin{aligned} p(\beta) &\sim \text{Gama}(1 + \beta|a, b) = \\ &= \frac{1}{\Gamma[a]b^a} (1 + \beta)^{a-1} \exp \left[ -\frac{(1 + \beta)}{b} \right], \end{aligned} \quad (3.26)$$

onde  $a$  e  $b$  são parâmetros ajustáveis. Foi usado  $a = b = 2$ , que dá uma distribuição *a priori* com 95% da densidade dentro da faixa  $[-\frac{1}{2}, \frac{21}{2}]$ , como mostra a Figura 9. A estimação pode ser realizada utilizando algum método de gradiente, ou empregando algum outro método de otimização convexa.

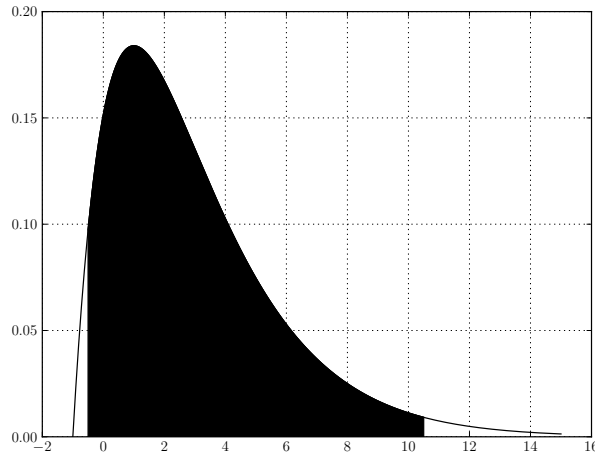


Figura 9 – Distribuição *a priori* de  $\beta$ .

### 3.3 Modificação Proposta

O algoritmo ICA-MAP(gen) pode ser implementado através do método do gradiente, como sugerido e descrito em (ZOU *et al.*, 2008). Porém o método não estabiliza devido à

algumas características das pdf's modeladas. A modelagem das pdf's é realizada no espaço da transformação ICA linear, utilizando o GGM.

Em (CHICHOCKI *et al.*, 2004), observa-se que, em muitas aplicações, é preciso realizar algum tipo de transformação nos sinais observados a fim de salientar alguma característica especial dos dados. Na ICA, os dados se tornam mais esparsos no espaço transformado. Neste caso, representação esparsa significa que o sinal transformado contém maior parte da sua energia em poucas amostras, as demais são próximas de zero. Em geral, esparsidade e independência estatística são critérios completamente diferentes, mas para variáveis aleatórias distribuídas com densidade gaussiana generalizada, que é o caso da GGM, os critérios de esparsidade e *kurtosis* normalizada coincidem para sinais supergaussianos, com *kurtosis* positiva. Pode-se ver na Figura 8 que a medida que o valor do parâmetro  $\beta$  do GGM aumenta, o pico em zero e a cauda da densidade aumentam, ou seja, a densidade dos dados se torna mais esparsa.

Devido à característica esparsa dos dados transformados, a convergência do algoritmo em (3.5) é dificultada. Quando o parâmetro  $\beta$  do GGM é maior que 1, a pdf apresenta um ponto crítico em zero, ou seja, a primeira derivada não existe. Ao derivar o GGM, este ponto em zero se torna descontínuo, como se pode ver no exemplo na Figura 10, utilizando o logaritmo do GGM com  $\beta = 2$ . Ou seja, a função não é diferenciável em zero.

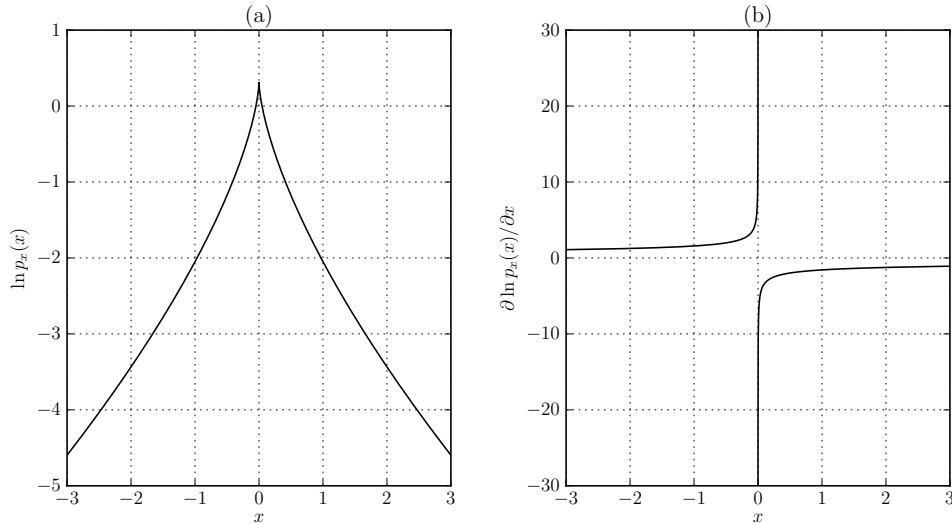


Figura 10 – (a) Logaritmo natural do GGM com  $\beta = 2$ . (b) Derivada de (a).

O método do gradiente ascendente (RAO, 2009), aplicado em (3.5), é um algoritmo de otimização utilizado para resolver o problema de maximização de funções contínuas,

$$\arg \max_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \quad (3.27)$$

onde  $f(\mathbf{x})$  é a função a ser maximizada. O algoritmo procura maximizar  $f(\mathbf{x})$  a partir de pequenos passos proporcionais ao gradiente de  $f(\mathbf{x})$  em algum ponto, sendo que o

gradiente retorna a direção de máximo. Com isso, o algoritmo básico é

$$\mathbf{x} \leftarrow \mathbf{x} + \lambda \nabla f, \quad (3.28)$$

onde  $\lambda$  é um escalar que ajusta o tamanho do passo a ser dado e  $\nabla f = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$  é o gradiente da função a ser maximizada.

O gradiente do algoritmo ICA-MAP(gen) é composto por funções como o da Figura 10(b), onde o gradiente retorna valores grandes próximos de zero. No caso do ICA-MAP(gen), a característica de esparsidade é desejável, mas traz essa dificuldade, pois em algum momento os valores de  $\mathbf{x}$  passarão em torno de zero, ou convergirão para valores próximos de zero. O ICA-MAP(gen) em (ZOU *et al.*, 2008) não explicita esse problema, e nem como é tratado. A Figura 11 mostra o exemplo da evolução de uma amostra em (3.5) e seu respectivo gradiente.

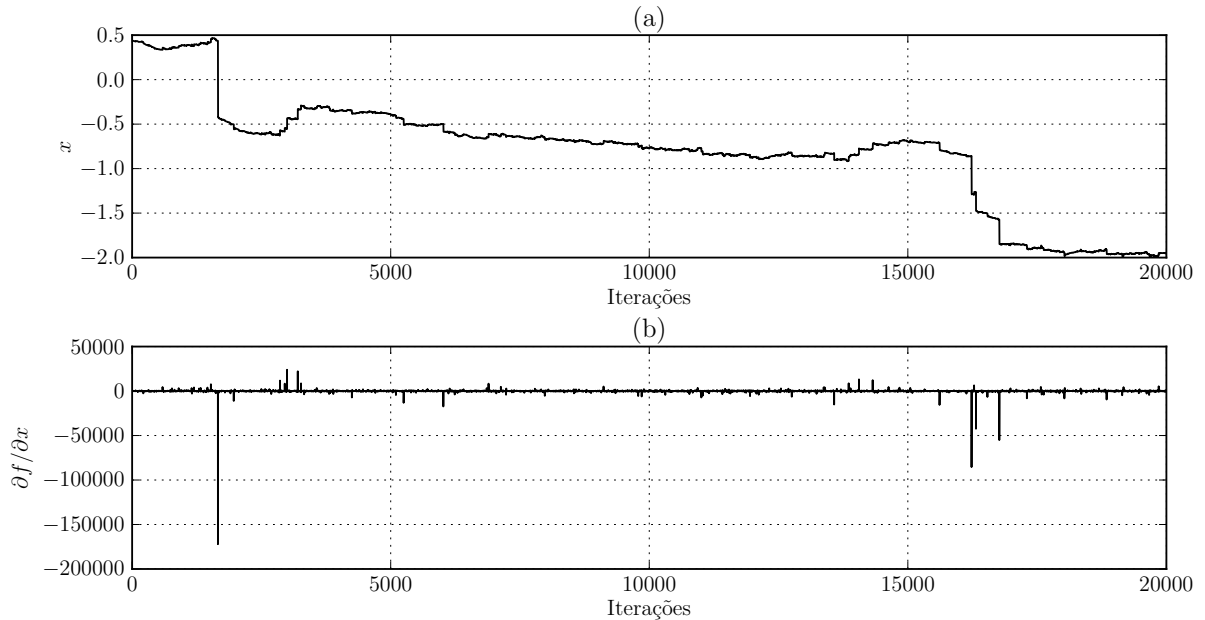


Figura 11 – (a) Evolução de uma componente de  $\mathbf{x}$  no algoritmo ICA-MAP(gen). (b) Evolução do gradiente em (3.5).

Pode-se ver que, em alguns instantes, o gradiente retorna um valor grande e faz o valor de  $x$  saltar inesperadamente. A fim de resolver este problema, é proposto algum tipo de limitação do gradiente. Através de alguns testes, verificou-se que uma solução é limitar a magnitude do vetor gradiente em um limiar  $L$ . Então, faz-se

$$l(g_i) = \begin{cases} g_i & \text{se } |g_i| < L \\ \text{sgn}(g_i)L & \text{caso contrário} \end{cases}, \quad (3.29)$$

onde  $g_i$  é cada elemento do vetor gradiente, e  $\text{sgn}(\cdot)$  é a função sinal. Então, a cada iteração, é verificado se a magnitude da  $i$ -ésima componente do vetor gradiente excede um certo

limiar  $L$ . A Figura 12 mostra a evolução da mesma amostra da Figura 11 adicionando a limitação em (3.29) com  $L = 400$ .

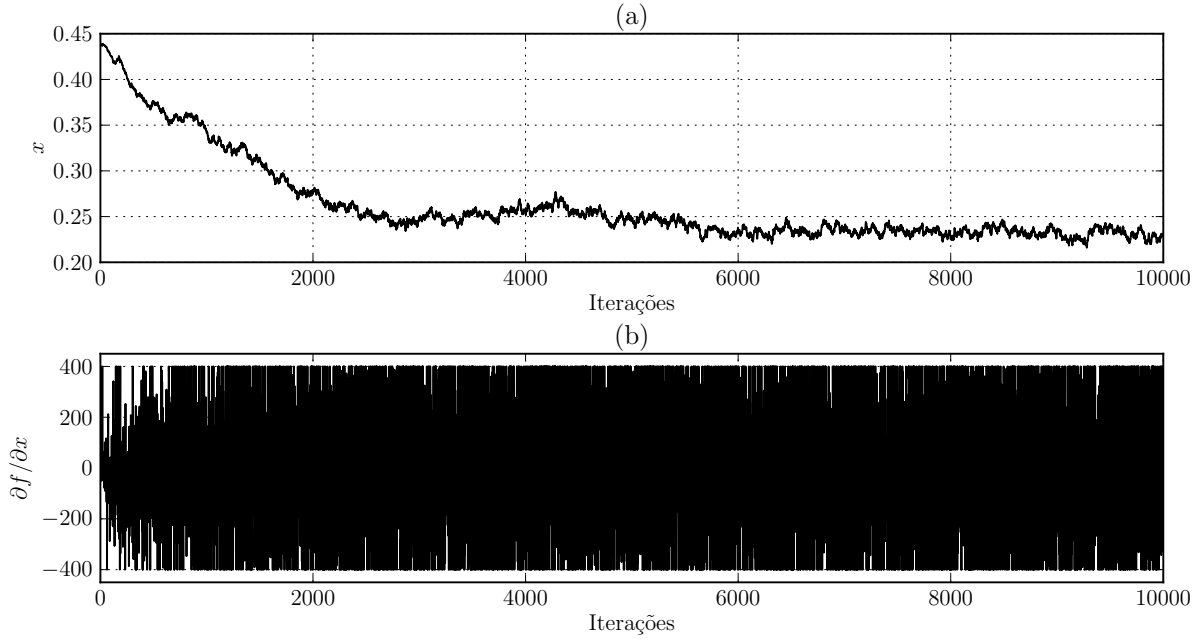


Figura 12 – (a) Evolução de uma componente de  $\mathbf{x}$  no algoritmo ICA-MAP(gen) com limitação do gradiente. (b) Evolução do gradiente em (3.5) com a limitação em (3.29).

Com essa modificação é possível alcançar a convergência do algoritmo e eliminar o problema das amostras que convergem para valores próximos de zero. Pode-se ainda entender a introdução dessa modificação como uma restrição ao problema de otimização. O algoritmo torna-se, então

$$\begin{aligned} \mathbf{x} &\leftarrow \arg \max_{\mathbf{x}} (\ln p_{s_v}(W^v(\mathbf{y} - \mathbf{x})) + \ln p_{s_x}(W^x \mathbf{x})) \\ &\text{com } \text{abs}(\nabla f) \leq L, \end{aligned} \quad (3.30)$$

onde  $f = \ln p_{s_v}(W^v(\mathbf{y} - \mathbf{x})) + \ln p_{s_x}(W^x \mathbf{x})$  é a função a ser maximizada, e  $\text{abs}(\cdot)$  é uma função que retorna o valor absoluto de cada componente de um vetor.

A fim de acelerar o algoritmo, é proposto utilizar um passo  $\lambda$  variável. Com a aplicação da limitação do gradiente, o valor máximo no vetor gradiente é  $L$ . Com isso, pode-se configurar um passo máximo para o algoritmo. Então, a cada iteração, recalcula-se o passo com

$$\lambda_{it} = \frac{1}{\max(\nabla f)} \lambda_0 \quad (3.31)$$

onde  $\max(\cdot)$  é uma função que retorna o valor máximo de um vetor, e  $\lambda_0$  é o parâmetro que configura o passo máximo. Então, o algoritmo torna-se

$$\mathbf{x} \leftarrow \mathbf{x} + \frac{1}{\max(\nabla f)} \lambda_0 \nabla f. \quad (3.32)$$



Observa-se que o tamanho do passo que atualiza  $\mathbf{x}$  é normalizada por  $\lambda_{it}$ . Dessa forma,  $\lambda_0$  é a magnitude do passo da componente de maior valor no vetor gradiente, e as demais são menores. A Figura 13 mostra um exemplo com a implementação do passo variável.

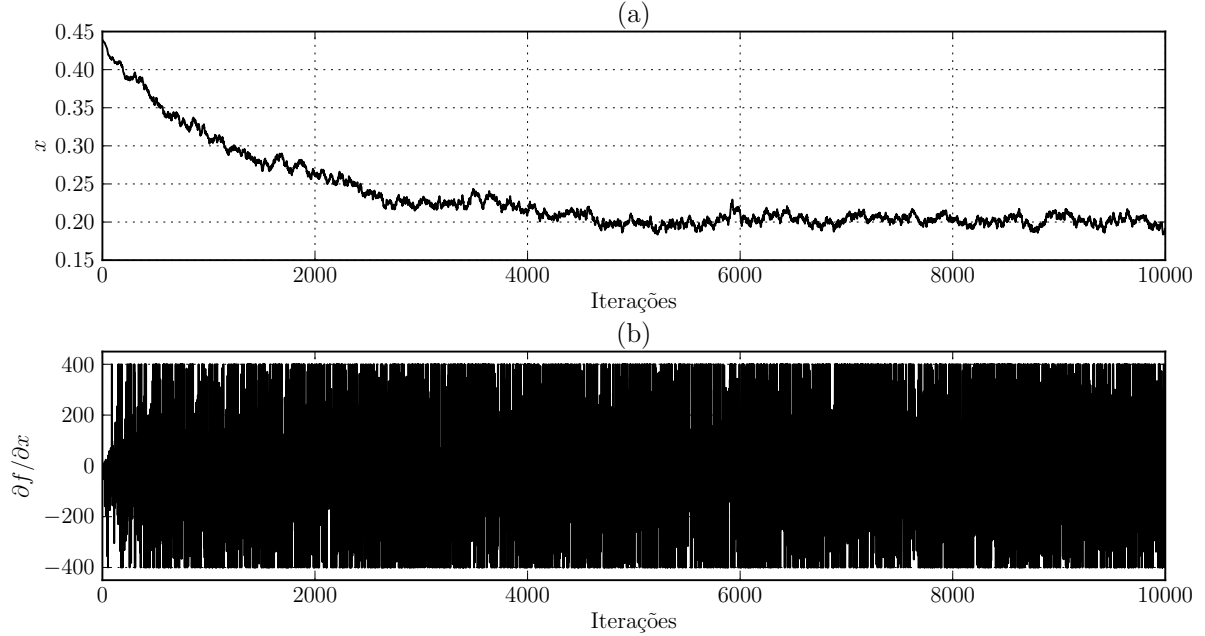


Figura 13 – (a) Evolução de uma componente de  $\mathbf{x}$  no algoritmo ICA-MAP(gen) com limitação do gradiente e com passo variável. (b) Evolução do gradiente em (3.5) com a limitação em (3.29) e passo variável em (3.31).

A vantagem do passo variável é a característica de configuração automática, ou seja, no começo do algoritmo, quando a solução está distante, é necessário ter um passo maior, e à medida que o algoritmo caminha para a solução o passo diminui, a fim de obter uma sintonia mais fina. Para ilustrar, a Figura 14 mostra o valor de  $\lambda_{it}$  calculado em cada iteração.

É interessante observar, nas Figuras 12 e 13, que nas proximidades do ponto de convergência do algoritmo o valor de gradiente alterna entre  $-L$  e  $L$ . Isso mostra que o problema da descontinuidade na derivada do GGM em Figura 10 é crítico. Então, uma abordagem alternativa à limitação do gradiente para amenizar este problema é aproximação polinomial nas proximidades de zero. A proposta é utilizar o *spline* Hermite cúbico em um intervalo único em torno de zero.

*Spline* Hermite cúbico é tipicamente usado para interpolar dados a fim de obter uma função contínua suave. A *spline* resultante deve ser contínua e ter primeira derivada contínua. No caso presente, é utilizado um intervalo único de interpolação na derivada do GGM próximo de zero. A interpolação de  $x$  em um intervalo arbitrário  $(x_k, x_{k+1})$  é

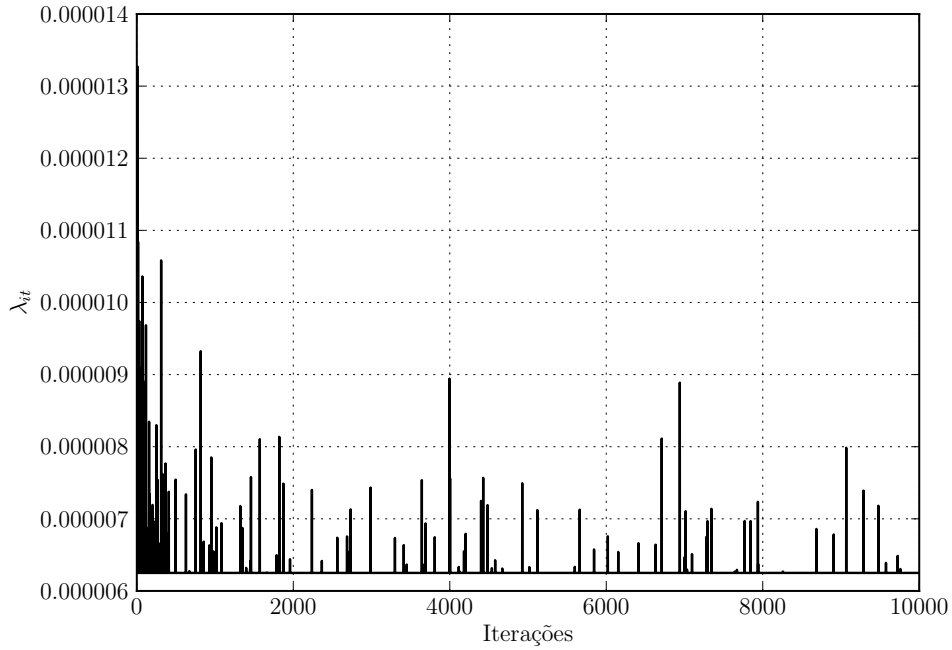


Figura 14 – Valores de  $\lambda_{it}$  calculados no decorrer das iterações usando limitação do gradiente.

realizado, através de

$$\begin{aligned}
 p(x) = & h_{00}(t)p_k + h_{10}(t)(x_{k+1} - x_k)m_k + \\
 & + h_{01}(t)p_{k+1} + h_{11}(t)(x_{k+1} - x_k)m_{k+1}, \\
 \text{com } t = & \frac{x - x_k}{x_{k+1} - x_k},
 \end{aligned} \tag{3.33}$$

onde  $p_k$  e  $p_{k+1}$  são os valores da função em  $x_k$  e  $x_{k+1}$ , respectivamente,  $m_k$  e  $m_{k+1}$  são as derivadas da função em  $x_k$  e  $x_{k+1}$ , respectivamente, e as funções  $h_{ij}$  são funções base definidas como

$$\begin{aligned}
 h_{00}(t) &= 2t^3 - 3t^2 + 1 \\
 h_{10}(t) &= t^3 - 2t^2 + t \\
 h_{01}(t) &= -2t^3 + 3t^2 \\
 h_{11}(t) &= t^3 - t^2.
 \end{aligned} \tag{3.34}$$

Ao aplicar a interpolação na derivada do GGM presente em Figura 10(b), o gradiente do algoritmo ICA-MAP(gen) será contínuo. A Figura 15 mostra um exemplo da interpolação. Assim como na limitação do gradiente, é necessário escolher quais são as fronteiras do intervalo de interpolação. As fronteiras do intervalo foram configuradas nos pontos onde  $\partial p_x(x)/\partial x = \pm 400$ , e é possível perceber isso na Figura 15.

Essa abordagem com *spline* evitará os valores grandes do gradiente quando o algoritmo ICA-MAP(gen) está próximo de zero, diferentemente da abordagem de limitação

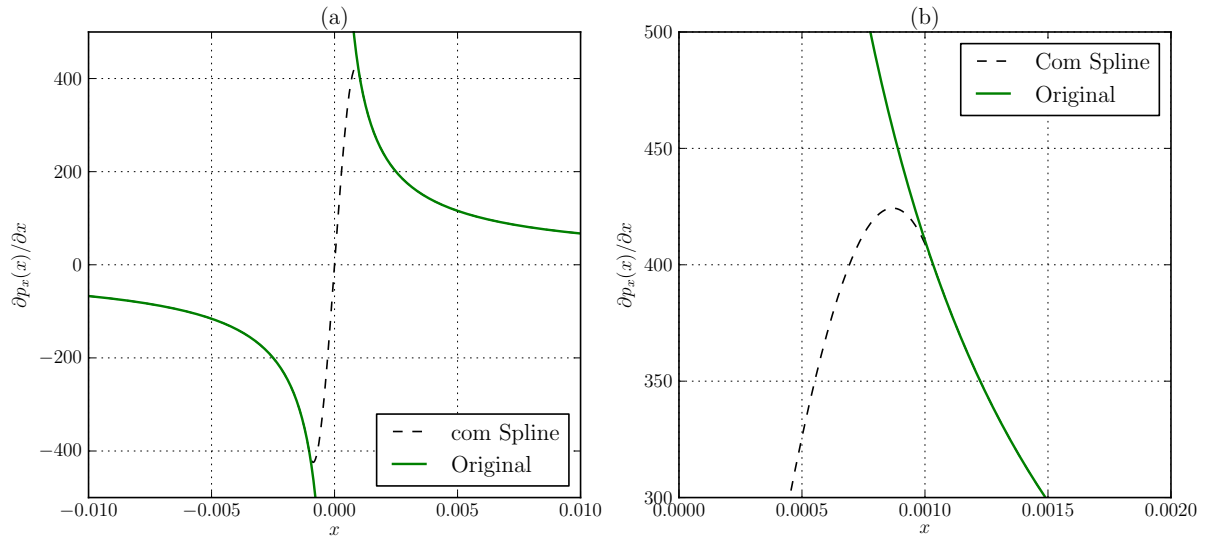


Figura 15 – (a) Exemplo de interpolação da derivada do GGM por *spline* Hermite cúbica em torno de zero. (b) *Zoom* de (a) na fronteira do intervalo de interpolação.

do gradiente, que alterna o valor do gradiente entre  $-L$  e  $L$  em torno de zero. Com isso, a evolução do algoritmo ICA-MAP(gen) se torna menos ruidosa, como mostra o exemplo da Figura 16.

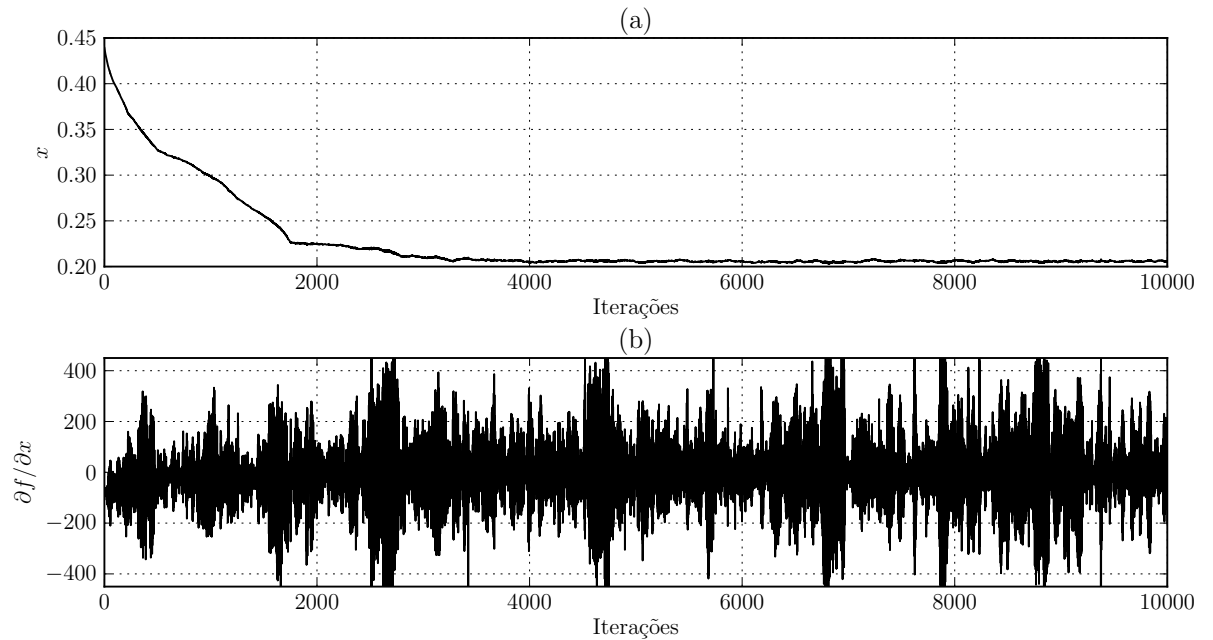


Figura 16 – (a) Evolução de uma componente de  $\mathbf{x}$  no algoritmo ICA-MAP(gen) com interpolação do gradiente por *spline*. (b) Evolução do gradiente em (3.5) com a interpolação do gradiente.

A adaptação do passo em (3.31) foi usada na abordagem de interpolação por *spline*

da Figura 16, e os valores calculados em cada iteração estão na Figura 17. Como no caso

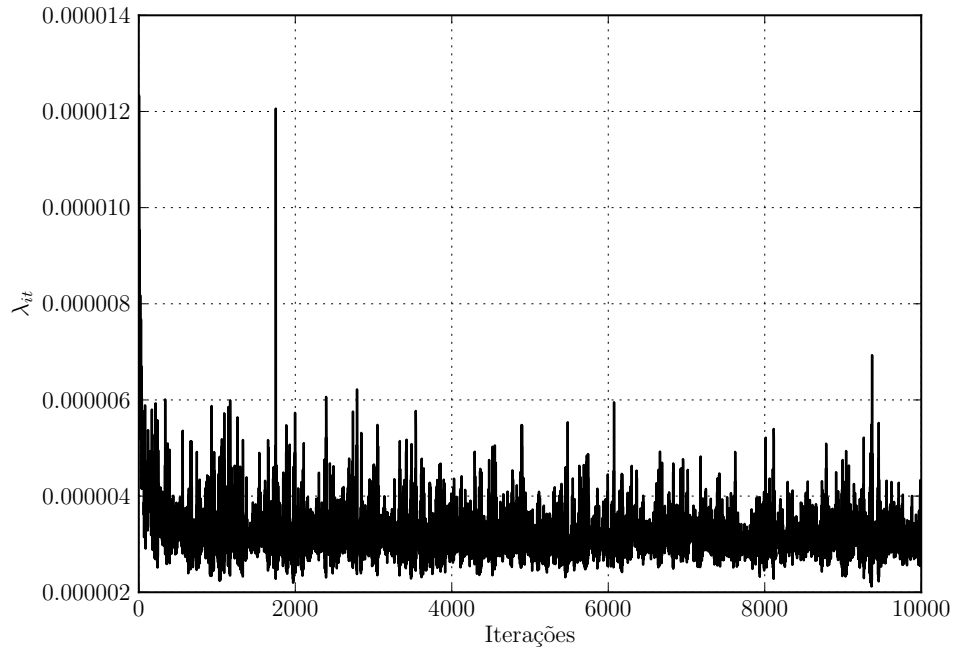


Figura 17 – Valores de  $\lambda_{it}$  calculados no decorrer das iterações usando interpolação por *spline*.

da limitação do gradiente, o passo se inicia com valores maiores e diminui ao decorrer das iterações. Neste trabalho, serão utilizadas as duas abordagens, limitação do gradiente e interpolação por *spline*, ambas com passo variável.

## 4 Testes e resultados

### 4.1 Metodologia para realização dos testes

O algoritmo ICA-MAP(gen) é composto de duas etapas. A primeira etapa é a de treino, onde são estimadas as matrizes de transformação  $W^x$  e  $W^v$ , e os parâmetros  $\beta$  do GGM para cada componente independente dos sinais de voz e ruídos. A segunda etapa é a de teste, onde os parâmetros encontrados na primeira etapa são usadas em (3.5) a fim de estimar o sinal limpo. Os experimentos foram realizados utilizando sinais de voz da base de dados TIMIT (GAROFALO, 1993)<sup>1</sup>, separados por gênero. Para o conjunto de treino de voz masculina foram escolhidos os arquivos

dr1-mcpm0/sa1    dr2-marc0/sa2    dr3-madc0/si1367    dr4-maeb0/si1411  
dr1-mcpm0/sx384    dr2-marc0/sx378    dr3-madc0/sx377    dr4-maeb0/sx90,

enquanto para o conjunto de treino de voz feminina foram usados os arquivos

dr1-fvmh0/sa1    dr2-faem0/sa2    dr3-falk0/si1086    dr4-falr0/si1325  
dr1-fvmh0/sx386    dr2-faem0/sx42    dr3-falk0/sx96    dr4-falr0/sx65.

Os arquivos foram concatenados, eliminando os silêncios entre as sentenças, dessa forma o treino é realizado apenas com a informação do sinal de voz. Na base TIMIT, existe um arquivo de extensão `.phn` para cada sinal de voz, que indica cada fonema bem como os silêncios. Tais arquivos foram usados num *script* para remover os instantes de silêncio. Os ruídos escolhidos para contaminar os sinais de voz são da base de dados DEMAND<sup>2</sup> e ISOLET<sup>3</sup>. Os tipos de ruídos escolhidos para realizar os testes foram `cafeter`, `f16`, `factory`, `station` e `gaussiano`. Os treinos de voz e ruído foram realizados usando 30 segundos de áudio. Todos os sinais de áudio das bases de dados, voz e ruído, estão amostrados originalmente em uma taxa de 16kHz, logo foi necessário reamostrar para 8kHz, que foi a taxa usada em (ZOU *et al.*, 2008). A reamostragem foi executada através do aplicativo *SoX Resampler Library*. Os treinos foram realizados através de estimativa MAP e a otimização através do método do gradiente. O tamanho do *frame* utilizado em (3.3) foi de  $N = 64$  amostras, e gerados usando janela retangular. O algoritmo fastICA utilizado para obter as transformações ICA é parte do pacote *MDP Toolkit*<sup>4</sup>.

Os arquivos utilizados para o conjunto de teste de voz masculina foram

<sup>1</sup> Amostras da base TIMIT disponíveis gratuitamente em <[http://nltk.github.com/nltk\\_data/packages/corpora/timit.zip](http://nltk.github.com/nltk_data/packages/corpora/timit.zip)>

<sup>2</sup> Disponível em <<http://parole.loria.fr/DEMAND/>>

<sup>3</sup> Disponível em <<http://www1.icsi.berkeley.edu/Speech/papers/eurospeech05-onset/isolet/>>

<sup>4</sup> Disponível em <<http://mdp-toolkit.sourceforge.net/>>

dr5-mbgt0/sx351 dr5-mbgt0/sx441 dr6-mbma1/sx324 dr6-mbma1/sx414  
 dr7-madd0/sx358 dr7-madd0/sx448 dr8-mbcg0/sx327 dr8-mbcg0/sx417,

e os arquivos do conjunto de teste para voz feminina foram

dr5-ftlg0/sx303 dr5-ftlg0/sx393 dr6-fapb0/sx343 dr6-fapb0/sx433  
 dr7-fblv0/sx338 dr7-fblv0/sx428 dr8-fbcg1/sx352 dr8-fbcg1/sx442.

É importante salientar que as sentenças dos locutores utilizadas para o treino não foram inseridas no conjunto de teste. E da mesma forma, os trechos de ruído utilizados para o treino não foram usados nos testes. Com (2.11), foi estimada a *kurtosis* normalizada de cada componente independente na etapa de treino. A Figura 18 mostra uma das estimativas, onde é possível perceber que todas as componentes são super-gaussianas (*kurtosis* positiva), portanto são esparsas. Todas as estimativas da *kurtosis* normalizada nos demais treinos acusaram super-gaussianidade, sem exceção.

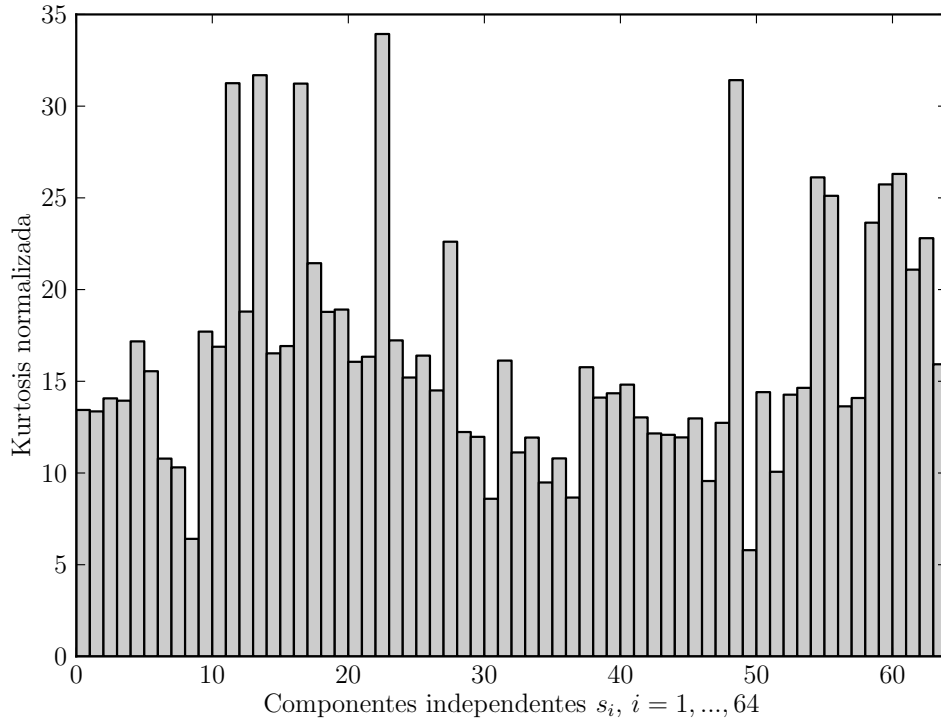


Figura 18 – Verificação da esparsidade das componentes independentes.

O algoritmo ICA-MAP(gen) em (3.5) foi testado com a limitação do gradiente em (3.29) com  $L = 500$  e interpolação com *spline* em (3.33) com  $p_0 = -500$  e  $p_1 = 500$ , e foi comparado com os algoritmos de subtração espectral (BOLL, 1979) (EPHRAIM; MALAH, 1984), por ser um método clássico, e *Sparse Code Shrinkage* (HYVÄRINEN, 1999) (POTAMITIS; FAKOTAKIS; KOKKINAKIS, 2001), por ser um caso especial do ICA-MAP. Na subtração espectral, foi usado 0,5 segundo para estimar o espectro do ruído.

Os algoritmos foram implementados usando a linguagem de programação Python

2.7, juntamente com as bibliotecas de computação científica Numpy, Scipy e Matplotlib. Alguns trechos do código foram implementados em Fortran 95 e integrados com o Python através do gerador de interface `f2py`. O uso de Fortran 95 foi necessário para acelerar o tempo de execução de alguns *loops* no código. O algoritmo ICA-MAP(gen) tem um custo computacional mais alto que *Sparse Code Shrinkage*, que utiliza uma solução de forma fechada. O ICA-MAP(gen) pode ser muito sensível aos parâmetros utilizados no processo de estimação.

Os testes serão avaliados por meio das medidas de relação sinal-ruído (SNR) e distorção espectral (SD). Denominando o sinal limpo como  $x$  e o sinal estimado como  $\hat{x}$ , o cálculo da SNR é

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \frac{\sum x^2}{\sum (x - \hat{x})^2}. \quad (4.1)$$

Apenas a informação da SNR não é suficiente para avaliar um algoritmo, então também é avaliada a SD. A distorção espectral entre  $x$  e  $\hat{x}$  é calculada através de

$$\text{SD} = \frac{1}{I} \sqrt{\sum_{i=0}^I \sum_{k=0}^{255} \left( 10 \left( \log_{10} \frac{|X(k, i)|}{|\hat{X}(k, i)|} \right) \right)^2}, \quad (4.2)$$

onde os sinais  $x$  e  $\hat{x}$  são previamente normalizados com  $x = x / \|x\|$  e  $\hat{x} = \hat{x} / \|\hat{x}\|$  para que a diferença de amplitude entre os sinais sejam desconsideradas,  $k$  e  $i$  são os índices de frequência e de *frame*, respectivamente, e  $|X(k, i)|$  e  $|\hat{X}(k, i)|$  são as amplitudes dos espectros de frequência de  $x$  e  $\hat{x}$ , respectivamente. A SD é uma medida de divergência entre espectros de frequências de dois sinais. Se os espectros são iguais, a SD é igual a zero, então é desejável que a SD seja a menor possível. Para o cálculo da SD, os sinais são divididos em *frames* não sobrepostos de 64 amostras. Entretanto, são adicionados 192 zeros em cada *frame* a fim de calcular a transformada rápida de Fourier (FFT) com uma janela de 256 amostras.

## 4.2 Testes realizados

Nos testes, os sinais de voz foram corrompidos por cada tipo de ruído a fim de que a SNR seja 0dB e 5dB. Os valores de SNR e SD calculados nos testes descritos estão presentes nas Tabelas 1 e 2. Como são 8 testes para cada caso, os valores tabelados são as médias de cada caso. Os resultados individuais podem ser encontrados na página <http://www2.ele.ufes.br/~evandro/denoising>. Os algoritmos ICAMAP 1 e ICAMAP 2 mencionados nas tabelas são os algoritmos ICA-MAP(gen) com limitação de gradiente e com aproximação por *spline*, respectivamente, ambos com implementação do passo variável. Não foi possível comparar com o ICA-MAP original pois a heurística utilizada para resolver o problema de convergência não foi descrita no artigo original.

Nos testes com ruído gaussiano em 0dB, observa-se que ICAMAP 1 e ICAMAP 2 possuem resultados semelhantes, com uma pequena vantagem para o ICAMAP 2. A Figura 19 mostra os gráficos de uma simulação de voz masculina contaminada com ruído gaussiano em 0dB. Pode-se ressaltar também que o ICA-MAP(gen) foi melhor que o *Sparse Code Shrinkage* (SCS), pois o SCS foi implementado com a aproximação de primeira ordem a fim de obter a solução fechada em (3.15). A consequência disso é o alto nível de resíduo ruidoso que o algoritmo deixa. Em termos de SNR, a subtração espectral teve um ganho maior, entretanto o ICA-MAP(gen) melhora bastante em termos de SD, ou seja, o espectro de frequência é mais próximo do ideal. No caso da subtração espectral, alguns artefatos são introduzidos no espectro, por isso que a SD piora bastante em alguns casos.

Já nos testes com ruído gaussiano em 5dB, o ICA-MAP(gen) se comporta melhor que os demais algoritmos em termos de SNR e SD. Nos demais algoritmos, o ganho de SNR é próximo do ICA-MAP(gen), mas em termos de SD, ICA-MAP(gen) é consideravelmente melhor. Esse resultado não foi atingido com SNR inicial de 0dB. Com isso, pode-se dizer que o ICA-MAP(gen) tem melhor desempenho em SNR's altas.

No caso de contaminação por ruídos não gaussianos em 0dB, pode-se ver na Tabela 1 que o ICA-MAP(gen) melhora bastante a SD. Já em termos de SNR, a melhoria não é grande. A Figura 20 mostra os gráficos de uma simulação de voz masculina contaminada com ruído não gaussiano *cafeter* em 0dB. A grande dificuldade em tratar de ruídos não gaussianos reais, como o barulho em uma estação de trem, pessoas falando em uma cafeteria, etc., é que, muitas vezes, eles são não estacionários. Os testes executados neste trabalho consideram estacionariedade dos sinais de áudio, onde a etapa de treinamento para cada tipo de ruído não é refeita no decorrer da estimação, e essa dificuldade se reflete nos resultados. Os demais algoritmos não se comportam bem. Eles introduzem alguns artefatos no espectro de frequência, e com isso, muitas vezes, pioram a SD.

Como ICA-MAP(gen) se comporta melhor para SNR maior, os testes com contaminação por ruído não gaussiano em 5dB têm bons resultados tanto em termos de SNR quanto de SD. Para ruídos não gaussianos, as demais técnicas não se comportam bem. Como no caso de SNR inicial em 0dB, isso é esperado, pois foram originalmente desenvolvidos para tratar de ruídos gaussianos.



Tabela 1 – Comparação de SNR (em dB) e SD de sinais corrompidos com 0 dB por ruídos gaussianos e não-gaussianos.

	Homem				Mulher			
	SNR inicial	SNR final	SD inicial	SD final	SNR inicial	SNR final	SD inicial	SD final
Gaussiano								
Subtração Espectral	0,00	5,41	9,82	6,61	0,00	6,17	9,48	6,36
Sparse Code Shrinkage	0,00	4,96	9,81	8,46	0,00	5,19	9,46	8,37
ICAMAP 1	0,00	5,68	9,81	5,68	0,00	6,16	9,49	5,15
ICAMAP 2	0,00	<b>5,73</b>	9,81	<b>5,65</b>	0,00	<b>6,30</b>	9,47	<b>5,13</b>
Cafeter								
Subtração Espectral	0,00	<b>4,88</b>	7,17	7,56	0,00	<b>5,58</b>	6,69	9,55
Sparse Code Shrinkage	0,00	2,55	7,17	9,15	0,00	2,77	6,69	9,23
ICAMAP 1	0,00	4,14	7,17	6,51	0,00	5,00	6,69	6,01
ICAMAP 2	0,00	4,21	7,17	<b>6,41</b>	0,00	5,02	6,69	<b>5,93</b>
F16								
Subtração Espectral	0,00	<b>5,51</b>	8,12	6,55	0,00	<b>6,03</b>	7,73	6,20
Sparse Code Shrinkage	0,00	3,47	8,12	8,00	0,00	3,71	7,73	7,45
ICAMAP 1	0,00	4,82	8,12	6,08	0,00	5,58	7,73	5,59
ICAMAP 2	0,00	4,93	8,12	<b>6,06</b>	0,00	5,76	7,73	<b>5,58</b>
Station								
Subtração Espectral	0,00	<b>6,19</b>	7,07	6,90	0,00	<b>7,23</b>	6,70	6,06
Sparse Code Shrinkage	0,00	3,07	7,07	9,22	0,00	3,37	6,70	7,94
ICAMAP 1	0,00	5,05	7,07	6,22	0,00	6,19	6,70	5,76
ICAMAP 2	0,00	5,19	7,07	<b>6,14</b>	0,00	6,59	6,70	<b>5,65</b>
Factory								
Subtração Espectral	0,00	<b>4,72</b>	8,27	15,91	0,00	<b>5,27</b>	7,91	15,42
Sparse Code Shrinkage	0,00	2,96	8,27	8,72	0,00	3,31	7,91	7,43
ICAMAP 1	0,00	3,70	8,27	6,14	0,00	4,81	7,91	5,69
ICAMAP 2	0,00	3,75	8,27	<b>6,10</b>	0,00	4,93	7,91	<b>5,67</b>

Tabela 2 – Comparação de SNR (em dB) e SD de sinais corrompidos com 5 dB por ruídos gaussianos e não-gaussianos.

	Homem				Mulher			
	SNR inicial	SNR final	SD inicial	SD final	SNR inicial	SNR final	SD inicial	SD final
Gaussiano								
Subtração Espectral	5,0	7,53	8,84	6,67	5,0	8,39	8,52	6,45
Sparse Code Shrinkage	5,0	8,70	8,82	7,51	5,0	8,88	8,51	7,32
ICAMAP 1	5,0	9,29	8,82	<b>5,08</b>	5,0	9,45	8,51	<b>4,74</b>
ICAMAP 2	5,0	<b>9,40</b>	8,83	5,17	5,0	<b>9,61</b>	8,51	4,87
Cafeter								
Subtração Espectral	5,0	7,35	6,36	<b>5,58</b>	5,0	8,06	5,90	7,92
Sparse Code Shrinkage	5,0	6,74	6,36	7,67	5,0	6,91	5,90	6,54
ICAMAP 1	5,0	7,68	6,36	5,90	5,0	8,41	5,90	5,34
ICAMAP 2	5,0	<b>7,73</b>	6,36	5,78	5,0	<b>8,48</b>	5,90	<b>5,24</b>
F16								
Subtração Espectral	5,0	7,88	7,25	5,96	5,0	8,41	6,87	5,55
Sparse Code Shrinkage	5,0	7,49	7,25	6,81	5,0	7,68	6,87	6,16
ICAMAP 1	5,0	8,20	7,25	5,35	5,0	8,78	6,87	4,90
ICAMAP 2	5,0	<b>8,31</b>	7,25	<b>5,26</b>	5,0	<b>8,98</b>	6,87	<b>4,79</b>
Station								
Subtração Espectral	5,0	8,57	6,26	5,94	5,0	9,51	5,90	5,31
Sparse Code Shrinkage	5,0	7,17	6,26	6,86	5,0	7,44	5,90	6,50
ICAMAP 1	5,0	8,52	6,26	5,32	5,0	9,79	5,90	4,83
ICAMAP 2	5,0	<b>8,68</b>	6,26	<b>5,17</b>	5,0	<b>9,99</b>	5,90	<b>4,71</b>
Factory								
Subtração Espectral	5,0	6,96	7,37	13,93	5,0	7,53	7,03	12,09
Sparse Code Shrinkage	5,0	7,04	7,37	7,37	5,0	7,33	7,03	6,60
ICAMAP 1	5,0	6,97	7,37	5,62	5,0	7,86	7,03	5,19
ICAMAP 2	5,0	<b>7,06</b>	7,37	<b>5,48</b>	5,0	<b>8,05</b>	7,03	<b>5,04</b>

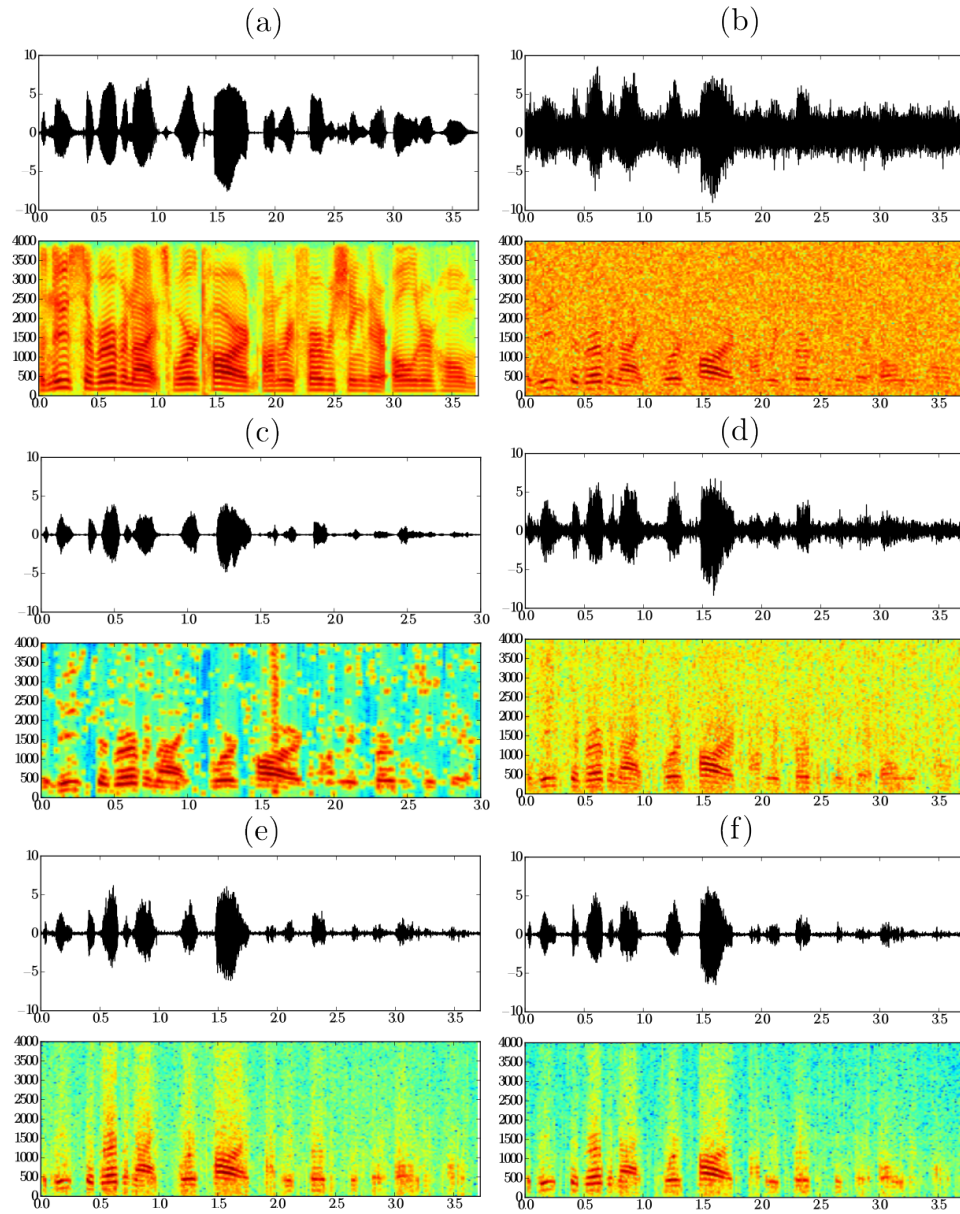


Figura 19 – (a) Sinal de voz masculina limpo e seu espectrograma. (b) Sinal contaminado com ruído gaussiano em 0 dB e seu espectrograma. (c) Estimativa do sinal limpo com subtração espectral. (d) Estimativa do sinal limpo com *Sparse Code Shrinkage*. (e) Estimativa do sinal limpo com ICAMAP 1. (f) Estimativa do sinal limpo com ICAMAP 2.

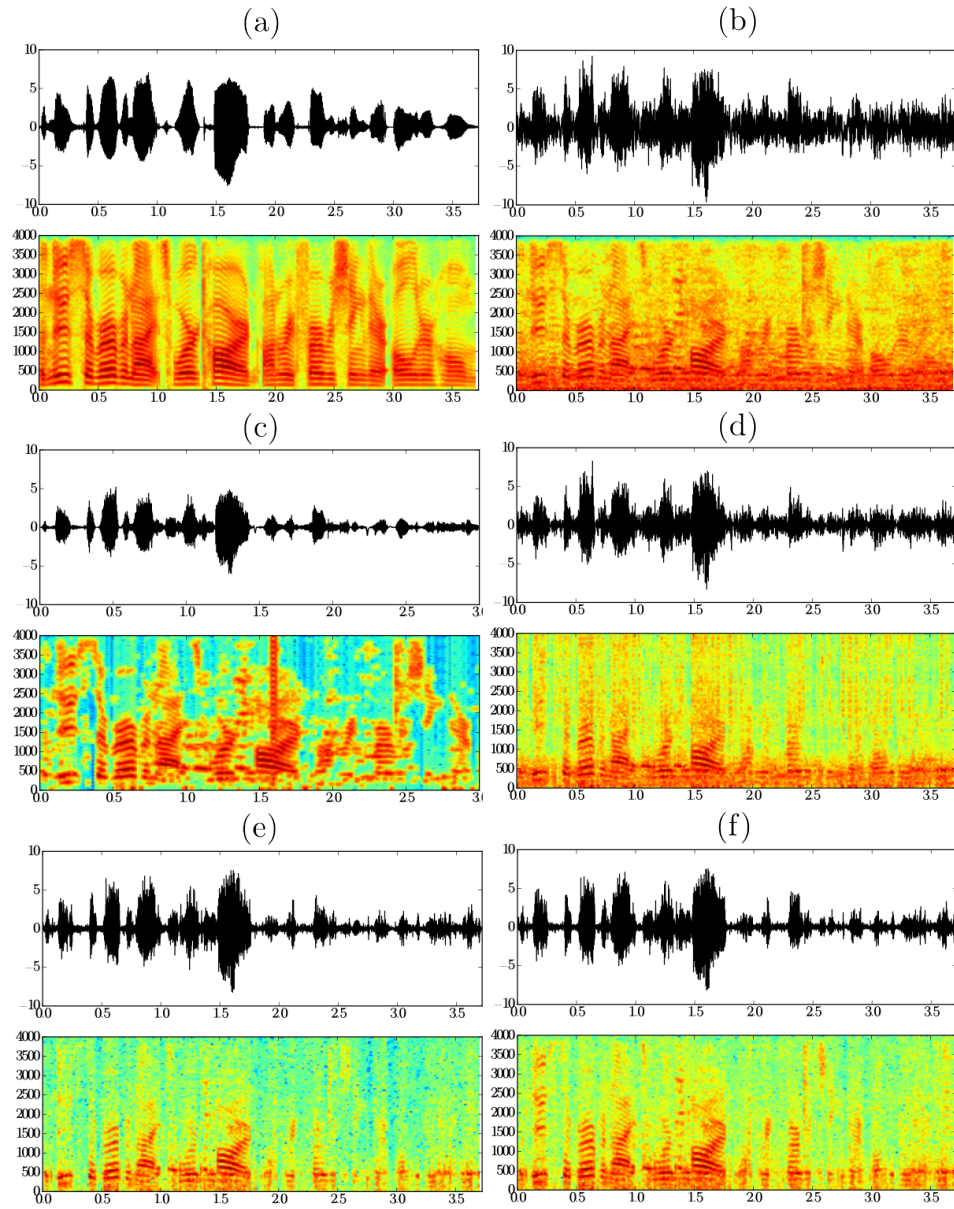


Figura 20 – (a) Sinal de voz masculina limpo e seu espectrograma. (b) Sinal contaminado com ruído não gaussiano em 0 dB e seu espectrograma. (c) Estimativa do sinal limpo com subtração espectral. (d) Estimativa do sinal limpo com *Sparse Code Shrinkage*. (e) Estimativa do sinal limpo com ICAMAP 1. (f) Estimativa do sinal limpo com ICAMAP 2.

## 5 Conclusões e Trabalhos Futuros

Neste trabalho foram apresentadas duas alterações na implementação do algoritmo ICA-MAP. A primeira abordagem é a limitação do vetor gradiente no algoritmo de otimização gradiente ascendente. Com isso, foi possível estabilizar a evolução do algoritmo. A segunda abordagem aplica uma interpolação através da *spline* Hermite cúbico, na derivada do modelo gaussiano generalizado. A interpolação é realizada em torno de zero, onde existe uma descontinuidade devido às características de esparsidade dos dados. Observou-se que o algoritmo ICA-MAP sempre melhora os sinais de voz, tanto em termos de relação sinal-ruído quanto da distorção espectral, independentemente da modificação implementada. Já os algoritmos de subtração espectral e *Sparse Code Shrinkage*, em alguns casos distorcem o espectro de frequência do sinal. Ao comparar as abordagens propostas, verificou-se que ambas tem comportamento semelhante. Notou-se que o ICA-MAP atingiu melhores resultados quando testado com SNR inicial de 5dB, podendo-se concluir que o algoritmo tem melhor desempenho em SNR's altas.

É importante salientar que o ICA-MAP depende de muitos fatores para ser bem executado. Como foi dito, o algoritmo é sensível aos parâmetros utilizados no processo de estimação. Com isso, é necessária uma boa estimativa das matrizes de separação do sinal limpo e do ruído, que levam os sinais para um domínio esparsos, e um bom ajuste dos modelos das *pdf*'s do sinal limpo e do ruído, que devem ser uma boa representação desses dados esparsos. Sabe-se que os critérios de independência estatística e esparsidade são compatíveis para distribuições super-gaussianas. Então, através da técnica ICA, é possível obter uma representação esparsa, onde o algoritmo maximiza as não-gaussianidades dos dados.

Para trabalhos futuros, uma sugestão é utilizar uma técnica alternativa onde seria possível obter uma transformação onde os dados transformados são mais esparsos que os encontrados na ICA. Uma possível técnica a ser utilizada é o algoritmo de otimização com norma  $\ell_1$ , ou fazer experimentos com ICA não-linear.

## Referências

BOLL, S. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, IEEE, v. 27, n. 2, p. 113–120, 1979. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1163209](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1163209)>. Citado 2 vezes nas páginas 13 e 45.

BOX, G. E.; TIAO, G. C. *Bayesian inference in statistical analysis*. [S.l.]: John Wiley & Sons, 1973. Citado na página 14.

BREITHAUPT, C.; MARTIN, R. Mmse estimation of magnitude-squared dft coefficients with supergaussian priors. In: IEEE. *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on*. 2003. v. 1, p. I–896. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1198926](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1198926)>. Citado na página 14.

CARDOSO, J.-F. Infomax and maximum likelihood for blind source separation. *Signal Processing Letters, IEEE*, IEEE, v. 4, n. 4, p. 112–114, 1997. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=566704](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=566704)>. Citado na página 34.

CARDOSO, J.-F. High-order contrasts for independent component analysis. *Neural computation*, MIT Press, v. 11, n. 1, p. 157–192, 1999. Disponível em: <<http://www.mitpressjournals.org/doi/abs/10.1162/089976699300016863>>. Citado na página 34.

CHICHOCKI, A.; LI, Y.; GEORGIEV, P.; AMARI, S.-i. Beyond ica: Robust sparse signal representations. In: IEEE. *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*. 2004. v. 5, p. V–684. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1329900](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1329900)>. Citado na página 37.

COLLINS, G. *Fundamental numerical methods and data analysis*. [S.l.: s.n.], 1990. Citado na página 33.

DELFOSSÉ, N.; LOUBATON, P. Adaptive blind separation of independent sources: a deflation approach. *Signal processing*, Elsevier, v. 45, n. 1, p. 59–83, 1995. Disponível em: <<http://www.sciencedirect.com/science/article/pii/016516849500042C>>. Citado na página 27.

EPHRAIM, Y.; MALAH, D. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 32, n. 6, p. 1109–1121, 1984. ISSN 0096-3518. Citado 2 vezes nas páginas 13 e 45.

FUKUNAGA, K. *Introduction to Statistical Pattern Recognition*. Elsevier Science, 1990. (Computer science and scientific computing). ISBN 9780080478654. Disponível em: <<http://books.google.com.br/books?id=BIJZTGjTxBgC>>. Citado 4 vezes nas páginas 8, 18, 19 e 20.

GAROFALO, J. S. *TIMIT: acoustic-phonetic continuous speech corpus*. [S.l.]: Linguistic Data Consortium, 1993. Citado na página 44.



GAZOR, S.; ZHANG, W. Speech enhancement employing laplacian-gaussian mixture. *Speech and Audio Processing, IEEE Transactions on*, IEEE, v. 13, n. 5, p. 896–904, 2005. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1495472](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1495472)>. Citado 2 vezes nas páginas 13 e 14.

HONG, L.; ROSCA, J.; BALAN, R. Bayesian single channel speech enhancement exploiting sparseness in the ica domain. In: *Proceedings of the EUSIPCO 2004*. [S.l.: s.n.], 2004. Citado 3 vezes nas páginas 13, 14 e 33.

HYVARINEN, A. A family of fixed-point algorithms for independent component analysis. In: IEEE. *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*. 1997. v. 5, p. 3917–3920. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=604766](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=604766)>. Citado na página 28.

HYVÄRINEN, A. Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural computation*, MIT Press, v. 11, n. 7, p. 1739–1768, 1999. Disponível em: <<http://www.mitpressjournals.org/doi/abs/10.1162/089976699300016214>>. Citado 4 vezes nas páginas 13, 14, 33 e 45.

HYVÄRINEN, A.; HOYER, P.; OJA, E. Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation. In: *Advances in Neural Information Processing Systems 11 (NIPS\*98)*. [S.l.]: MIT Press, 1999. p. 473–479. Citado na página 14.

HYVÄRINEN, A.; KARHUNEN, J.; OJA, E. *Independent Component Analysis*. Wiley, 2004. (Adaptive and Learning Systems for Signal Processing, Communications and Control Series). ISBN 9780471464198. Disponível em: <<http://books.google.com.br/books?id=96D0ypDwAkkC>>. Citado 3 vezes nas páginas 21, 22 e 23.

HYVARINEN, A.; OJA, E. A neuron that learns to separate one signal from a mixture of independent sources. In: IEEE. *Neural Networks, 1996., IEEE International Conference on*. 1996. v. 1, p. 62–67. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=548867](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=548867)>. Citado na página 27.

HYVÄRINEN, A.; OJA, E. A fast fixed-point algorithm for independent component analysis. *Neural computation*, MIT Press, v. 9, n. 7, p. 1483–1492, 1997. Disponível em: <<http://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.7.1483>>. Citado 2 vezes nas páginas 27 e 28.

JEANNES, R. L. B.; FAUCON, G. Proposal of a voice activity detector for noise reduction. *Electronics Letters, IET*, v. 30, n. 12, p. 930–932, 1994. Disponível em: <[http://digital-library.theiet.org/content/journals/10.1049/el\\_19940625](http://digital-library.theiet.org/content/journals/10.1049/el_19940625)>. Citado na página 14.

LEE, J.-H.; JUNG, H.-Y.; LEE, T.-W.; LEE, S.-Y. Speech enhancement with map estimation and ica-based speech features. Citeseer, 2000. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.8128>>. Citado na página 33.

LEE, T.-W.; LEWICKI, M. S. The generalized gaussian mixture model using ica. In: *International Workshop on Independent Component Analysis (ICA2000)*. [S.l.: s.n.], 2000. p. 239–244. Citado 2 vezes nas páginas 13 e 35.

- LIM, J. S.; OPPENHEIM, A. V. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, IEEE, v. 67, n. 12, p. 1586–1604, 1979. Disponível em: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1455809](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1455809). Citado na página 13.
- MILLER, S.; CHILDERS, D. *Probability and random processes: with applications to signal processing and communications*. [S.l.]: Access Online via Elsevier, 2004. Citado na página 24.
- POTAMITIS, I.; FAKOTAKIS, N.; KOKKINAKIS, G. Speech enhancement using the sparse code shrinkage technique. In: IEEE. *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP'01). 2001 IEEE International Conference on*. 2001. v. 1, p. 621–624. Disponível em: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=940908](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=940908). Citado 3 vezes nas páginas 13, 33 e 45.
- RAMIREZ, J.; SEGURA, J. C.; BENITEZ, C.; TORRE, A. D. L.; RUBIO, A. Efficient voice activity detection algorithms using long-term speech information. *Speech communication*, Elsevier, v. 42, n. 3, p. 271–287, 2004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167639303001201>. Citado na página 14.
- RAO, S. *Engineering Optimization: Theory and Practice*. Wiley, 2009. ISBN 9780470183526. Disponível em: <http://books.google.com.br/books?id=YNt34dvnQLEC>. Citado na página 37.
- SOON, I.; KOH, S. Low distortion speech enhancement. *IEE Proceedings-Vision, Image and Signal Processing*, IET, v. 147, n. 3, p. 247–253, 2000. Disponível em: [http://digital-library.theiet.org/content/journals/10.1049/ip-vis\\_20000323](http://digital-library.theiet.org/content/journals/10.1049/ip-vis_20000323). Citado na página 13.
- XIE, H.; PIERCE, L. E.; ULABY, F. T. Sar speckle reduction using wavelet denoising and markov random field modeling. *Geoscience and Remote Sensing, IEEE Transactions on*, IEEE, v. 40, n. 10, p. 2196–2212, 2002. Disponível em: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1105905](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1105905). Citado 2 vezes nas páginas 13 e 14.
- YOST, W. A.; POPPER, A. N.; FAY, R. R. *Auditory perception of sound sources*. [S.l.]: Springer, 2008. Citado na página 17.
- ZOU, X.; JANCOVIC, P.; KOKUER, M. On the effectiveness of the ica-based signal representation in non-gaussian noise. In: *Signal Processing, 2008. ICSP 2008. 9th International Conference on*. [S.l.: s.n.], 2008. p. 1–4. Citado na página 31.
- ZOU, X.; JANCOVIC, P.; LIU, J.; KOKUER, M. Speech signal enhancement based on map algorithm in the ica space. *Signal Processing, IEEE Transactions on*, v. 56, n. 5, p. 1812–1820, 2008. ISSN 1053-587X. Citado 7 vezes nas páginas 14, 30, 33, 34, 36, 38 e 44.